EMT-16

প্রাক্‌কথন

নেতাজি সুভাষ মুক্ত বিশ্ববিদ্যালয়ের স্নাতক শ্রেণির জন্য যে পাঠক্রম প্রবর্তিত হয়েছে, তার লক্ষণীয় বৈশিষ্ট্য হ'ল প্রতিটি শিক্ষার্থীকে তাঁর পছন্দমতো কোনও বিষয়ে সাম্মানিক (honours) স্তরে শিক্ষাগ্রহণের সুযোগ করে দেওয়া। এক্ষেত্রে ব্যক্তিগতভাবে তাঁদের গ্রহণক্ষমতা আগে থেকেই অনুমান করে না নিয়ে নিয়ত মূল্যায়ণের মধ্য দিয়ে সেটা স্থির করাই যুক্তিযুক্ত। সেই অনুযায়ী একাধিক বিষয়ে সাম্মানিক মানের পাঠ-উপকরণ রচিত হয়েছে ও হচ্ছে — যার মূল কাঠামো স্থিরীকৃত হয়েছে একটি সুচিন্তিত পাঠক্রমের ভিত্তিতে। কেন্দ্র ও রাজ্যের অগ্রগণ্য বিশ্ববিদ্যালয়সমূহের পাঠক্রম অনুসরণ করে তার আদর্শ উপকরণগুলির সমন্বয়ে রচিত হয়েছে এই পাঠক্রম। সেই সঙ্গে যুক্ত হয়েছে অধ্যেতব্য বিষয়ে নতুন তথ্য, মনন ও বিশ্লেষণের সমাবেশ।

দূর-সঞ্চারী শিক্ষাদানের স্বীকৃত পদ্ধতি অনুসরণ করেই এই সব পাঠ-উপকরণ লেখার কাজ চলছে। বিভিন্ন বিষয়ের অভিজ্ঞ পণ্ডিতমণ্ডলীর সাহায্য এ কাজে অপরিহার্য এবং যাঁদের নিরলস পরিশ্রমে লেখা, সম্পাদনা তথা বিন্যাসকর্ম সুসম্পন্ন হচ্ছে তাঁরা সকলেই ধন্যবাদের পাত্র। আসলে, এঁরা সকলেই অলক্ষ্যে থেকে দূরসঞ্চারী শিক্ষাদানের কার্যক্রমে অংশ নিচ্ছেন ; যখনই কোনো শিক্ষার্থী এই পাঠ্যবস্তুনিচয়ের সাহায্য নেবেন, তখনই তিনি কার্যত একাধিক শিক্ষকমণ্ডলীর পরোক্ষ অধ্যাপনার তাবৎ সুবিধা পেয়ে যাচ্ছেন।

এইসব পাঠ-উপকরণের চর্চা ও অনুশীলনে যতটা মনোনিবেশ করবেন কোনও শিক্ষার্থী, বিষয়ের গভীরে যাওয়া তাঁর পক্ষে ততই সহজ হবে। বিষয়বস্তু যাতে নিজের চেষ্টায় অধিগত হয়, পাঠ-উপকরণের ভাষা ও উপস্থাপনা তার উপযোগী করার দিকে সর্বস্তরে নজর রাখা হয়েছে। এরপর যেখানে যতটুকু অস্পষ্টতা দেখা দেবে, বিশ্ববিদ্যালয়ের বিভিন্ন পাঠকেন্দ্রে নিযুক্ত শিক্ষা-সহায়কগণের পরামর্শে তার নিরসন অবশ্যই হ'তে পারবে। তার ওপর প্রতি পর্যায়ের শেষে প্রদত্ত অনুশীলনী ও অতিরিক্ত জ্ঞান অর্জনের জন্য গ্রন্থ-নির্দেশ শিক্ষার্থীর গ্রহণ-ক্ষমতা ও চিন্তাশীলতা বৃদ্ধির সহায়ক হবে।

এই অভিনব আয়োজনের বেশ কিছু প্রয়াসই এখনও পরীক্ষামূলক—অনেক ক্ষেত্রে একেবারে প্রথম পদক্ষেপ। স্বভাবতই ত্রুটি-বিচ্যুতি কিছু কিছু থাকতে পারে, যা অবশ্যই সংশোধন ও পরিমার্জনার অপেক্ষা রাখে। সাধারণভাবে আশা করা যায়, ব্যাপকতর ব্যবহারের মধ্য দিয়ে পাঠ-উপকরণগুলি সর্বত্র সমাদৃত হবে।

অধ্যাপক (ড.) শুভ শঙ্কর সরকার

উপাচার্য

ষষ্ঠ পুনর্মুদ্রণ : নভেম্বর, 2019

বিশ্ববিদ্যালয় মঞ্জুরি কমিশনের দূরশিক্ষা ব্যুরোর বিধি অনুযায়ী ও অর্থানুকূল্যে মুদ্রিত।
Printed in accordance with the regulations and financial assistance of
the Distance Education Bureau of the University Grants Commission.

পরিচিতি

বিষয় : গণিত বিদ্যা **সাম্মানিক স্তর**

পাঠক্রম : পর্যায়
EMT 16 : 01

	রচনা	**সম্পাদনা**
একক 1-4	ড. সুমা দেবশর্মা	ড. মধুমঞ্জাল পাল
একক 5-6	অধ্যাপক সঞ্জয় সেন	ড. মধুমঞ্জাল পাল

প্রজ্ঞাপন

এই পাঠ-সংকলনের সমুদয় স্বত্ব নেতাজি সুভাষ মুক্ত বিশ্ববিদ্যালয়ের দ্বারা সংরক্ষিত। বিশ্ববিদ্যালয় কর্তৃপক্ষের লিখিত অনুমতি ছাড়া এর কোনোও অংশের পুনর্মুদ্রণ বা কোনোভাবে উদ্ধৃতি সম্পূর্ণ নিষিদ্ধ।

মোহন কুমার চট্টোপাধ্যায়
নিবন্ধক

নেতাজি সুভাষ মুক্ত বিশ্ববিদ্যালয়

EMT — 16

C ভাষার কম্পিউটারে নির্দেশলিপিকরণ
(স্নাতক পাঠক্রম)

পর্যায়
1

একক 1	❑	পরিগণকের প্রাথমিক পাঠ	7–24
একক 2	❑	তথ্য পরিবেশন	25–53
একক 3	❑	নির্দেশলিপিকরণ ভাষা	54–60
একক 4	❑	ক্রিয়ানুক্রম তালিকা	61–73
একক 5	❑	পরিগণকের ভাষা — C	74–96
একক 6	❑	C — নির্দেশাবলি	97–138

একক 1 ❑ পরিগণকের প্রাথমিক পাঠ

এককের উদ্দেশ্য ঃ

এই একক পাঠ করলে পাঠক নিম্নলিখিত বিষয়গুলি সম্বন্ধে জ্ঞাত হবে।

- **পরিগণক যন্ত্র কি?**
- **পরিগণক যন্ত্রের ঐতিহাসিক ক্রমবিকাশ**
- **পরিগণকের প্রকার ভেদ**
- **আধুনিক পরিগণক যন্ত্রের বিবরণ**
- **পরিগণক যন্ত্রের বিভিন্ন অংশের কার্যকারিতা**

পরিগণকের সাথে পরিচিতি ঃ

পরিগণক বা কম্পিউটারের নাম শোনেননি বা দেখেননি এমন মানুষ বর্তমানে খুব অল্পই খুঁজে পাওয়া যাবে। সমাজ জীবনের বিভিন্ন ক্ষেত্রে পরিগণক এখন এক অপরিহার্য অঙ্গ। বিভিন্ন শিল্প, শিক্ষা প্রতিষ্ঠান, গবেষণাকেন্দ্র, সরকারী-বেসরকারী কর্মক্ষেত্র, স্বাস্থ্যকেন্দ্র, ব্যাঙ্ক, রেল স্টেশন, বিমানবন্দর ইত্যাদি আরও বিভিন্ন কর্মক্ষেত্রে বৈপ্লবিক পরিবর্তন এনেছে পরিগণক নামক বৈদ্যুতিন্ যন্ত্রটি। শুধুমাত্র কর্মজীবনে নয়, বিনোদনের জগতেও পরিগণকের ব্যবহার হচ্ছে সৃজনশীল হাতিয়ার হিসাবে। সঙ্গীতের স্বরলিপি রচনা, চিত্রাঙ্কণ, দূরেক্ষণ খেলা ইত্যাদির চর্চা করাও সম্ভব বৈদ্যুতিন্ পরিগণকের মাধ্যমে। বৈদ্যুতিন প্রযুক্তিবিদ্যার আনুকুল্যে যে সমস্ত আবিষ্কারের সুফল দ্বারা জনজীবন প্রভাবিত তাদের মধ্যে বৈদ্যুতিন্ পরিগণক এককভাবে সবিশেষ উল্লেখযোগ্য স্থান অধিকার করে নিয়েছে। কিন্তু এর কারণ কি? একটি বৈদ্যুতিন্ পরিগণক কি করতে পারে? বর্তমান বৈদ্যুতিন্ পরিগণকের কিছু কর্মক্ষমতার কথা নীচে বলা হল—

(i) যোগ, বিয়োগ, গুণ, ভাগ এবং অন্য যে কোনও জটিল গাণিতিক সমস্যা সংক্রান্ত হিসাব নিকাশ করতে পারে পরিগণক একেবারে নির্ভুলভাবে এবং তড়িৎ-চকিত ক্ষিপ্রতায়।

(ii) বর্তমান বৈদ্যুতিন্ পরিগণকের সহায়তায় একই কাজ বারংবার করা যেতে পারে অনায়াসে অতি অল্প সময়ে এবং ক্লান্তিহীন ভাবে।

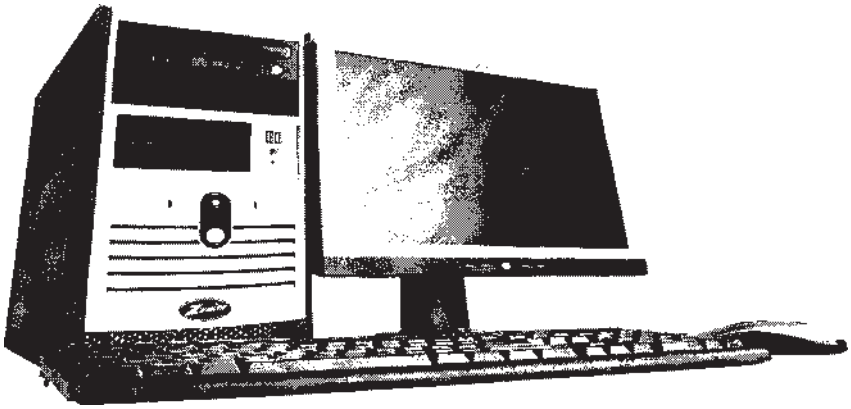
(iii) বর্তমান বৈদ্যুতিন্ পরিগণকের স্মৃতি কক্ষে/স্মৃতি আধারে সাংখ্যিক, অসাংখ্যিক বা মিশ্র প্রকৃতির তথ্য বা তথ্যসমৃদ্ধ বার্তা সঞ্চয় করে রাখার ক্ষমতা প্রায় অপরিসীম, তবে পরিমেয়।

(iv) বর্তমান বৈদ্যুতিন্ পরিগণক যুক্তিসম্মত উপায়ে তথ্য বাছাই, বিভিন্ন তথ্যের মধ্যে তুলনামূলক বিচার, কোন অপ্রয়োজনীয় তথ্যকে বিশেষ কোন তথ্য দ্বারা প্রতিস্থাপন ইত্যাদি যাবতীয় প্রয়োজনীয় তথ্যের সংসাধন সংক্রান্ত কাজ করতে সক্ষম।

(v) বর্তমান বৈদ্যুতিন্ পরিগণক তার ব্যবহারকারীর পরিকল্পনা রূপায়ণে এবং বিভিন্ন সমস্যা সমাধানে বিশেষ

গুরুত্বপূর্ণ স্থান অধিকার করে নিয়েছে। অনেক ক্ষেত্রে পরিগণক তার ব্যবহারকারীর নির্দেশাদি সম্পর্কে যথেষ্ট প্রতিক্রিয়াশীল।

বর্তমান বৈদ্যুতিন্ পরিগণকের এই সমস্ত কর্মদক্ষতা থেকে বলা যেতেই পারে যে, এই বৈদ্যুতিন্ যন্ত্রটির কার্যবিধি মানব মস্তিষ্কের মতই যুক্তিশীল। আর এই কারণেই যন্ত্রটির প্রয়োগক্ষেত্র সময়ের অগ্রগতির সাথে সাথে সুদূর প্রসারিত হয়েছে এবং হয়ে চলেছে। এমন দিন প্রায় এসেই গেছে যখন শৈশবে বর্ণমালা পাঠ আর পরিগণক পরিচিতি একই সাথে হবে ক্ষুদে ছাত্র-ছাত্রীদের!



চিত্র-১ ঃ বৈদ্যুতিন্ পরিগণক যন্ত্র

পরিগণক যন্ত্রের ঐতিহাসিক ক্রমবিকাশ ঃ

সভ্যতার ইতিহাসে গণনার ইতিহাস অতি সুপ্রাচীন। মানুষ কখন কোথায় প্রথম গণনা করতে শিখেছিল তা সুনির্দিষ্ট করে বলা যায় না। কিন্তু গণনার ইতিহাসে আমরা দেখতে পাই মানুষের নিরন্তর প্রয়াস—গণনার কাজকে নিয়মানুগ করে তোলার জন্য, জটিল থেকে জটিলতর গাণিতিক সমস্যাকে সহজ সুশৃঙ্খল উপায়ে দ্রুত এবং আরও দ্রুত সমাধান করার জন্য। আমরা যে দশমিক সংখ্যা প্রণালী ব্যবহার করি তার নিধান 10। হয়তো এর কারণ এটাই যে, দুই হাতের 10 টি অঙ্গুলির সাহায্যে গণনার সুবিধা মানুষ উপলব্ধি করতে পেরেছিল। গণনার কাজে গতি আনতে পারে এইরকম কোনও গণনা সহায়ক মাধ্যম বা যন্ত্র উদ্ভাবনের প্রচেষ্টা শুরু হয়েছিল সুদূর অতীতে। সবচেয়ে পুরনো গণনা সহায়ক জিনিসটির নাম অ্যাবাকাস (ABACUS)। প্রায় 3000 খ্রীষ্ট পূর্বাব্দে মধ্য প্রাচ্যে এবং পরবর্তীকালে চীন ও জাপানে বর্তমান অ্যাবাবাসের সমতুল্য একটি গণনা সহায়ক পরিকাঠামো ব্যবহারের উল্লেখ পাওয়া যায়। সবচেয়ে পুরনো এই গণনা মাধ্যমটি আজও ব্যবহার করা হয় ছোটদের মনে সংখ্যা ও যোগ-বিয়োগ সম্পর্কে স্বচ্ছ ধারণা তৈরী করতে। এছাড়া দৃষ্টিহীন ছাত্রছাত্রীদের পক্ষে এটি যথেষ্ট গ্রহণযোগ্য।

1614 সালে স্কটল্যাণ্ডের বিখ্যাত গণিতজ্ঞ জন ন্যাপিয়ের (John Napier) উদ্ভাবন করেন একটি গুণন প্রক্রিয়া সহায়ক মাধ্যম।

1633 সালে ন্যাপিয়ের-এর লগারিদম-তত্ত্বকে কাজে লাগিয়ে ইংরেজ গবেষক উইলিয়ম্ আউটরেড্ (William Oughtred) তৈরী করেছিলেন একটি গণনাসহায়ক জিনিস। এটির নাম ছিল আনুপাতিক বৃত্তমণ্ডলী (Circles of Proportion)। এর আধুনিকীকরণ হয় 1663 সালে এবং নতুন নাম হয় স্লাইড রুল (Slide Rule)। এটি তৈরী হত দুটি মাপনীর সাহায্যে। একটি মাপনী ছিল বিশেষভাবে চিহ্নিত এবং এটি খাঁজ বরাবর দ্রুত চলাচল করতে পারত। এর সাহায্যে যোগফল নির্ণয় করা যেত দ্রুত। মাত্র কয়েক দশক আগেও স্লাইড রুলের যথেষ্ট ব্যবহার ছিল।

1642 সালে ফরাসী দার্শনিক ও গণিতজ্ঞ ব্লেইসি পাস্কাল (Blaise Pascal) মাত্র ঊনিশ বছর বয়সে একটি গণনা সহায়ক যন্ত্র উদ্ভাবন করেছিলেন। এই যন্ত্রটির সাহায্যে দ্রুত যোগবিয়োগ এবং গুণ-ভাগ করা সম্ভব ছিল। পাস্কালের তৈরী এই যন্ত্রের ধারণাকে কাজে লাগিয়ে পরবর্তীকালে তৈরী হয়েছিল যান্ত্রিক টেবিল গাণনিক (Desk Calculator)।

স্বয়ংক্রিয় গণক যন্ত্র উদ্ভাবনের প্রচেষ্টা শুরু হয় ঊনবিংশ শতাব্দীর গোড়া থেকেই। জ্যাক্ উয়ার্ড, (Jacquard) নামে একজন গবেষক উদ্ভাবন করেন একটি স্বয়ংক্রিয় তাঁত যন্ত্র। যন্ত্রটি চালাতে হত সাংকেতিক ছিদ্র সমন্বিত কার্ড বা পান্শ্ কার্ড-এর সাহায্যে। প্রায় একই সময়ে 1823 সালে ইংরেজ গণিতজ্ঞ ও যন্ত্রশিল্পী চার্লস্ ব্যাবেজ্ (Charles Babage) একটি স্বয়ংক্রিয় গণক যন্ত্রের নক্সা তৈরী করেন। যন্ত্রটি তথ্য এবং নির্দেশ উভয়ই যাতে গ্রহণ করতে পারে সেই ভাবনা ছিল নক্সাটিতে। ব্যাবেজ্ এই যন্ত্রটির নাম দিয়েছিলেন ডিফারেন্স ইঞ্জিন (Difference Engine)। ব্যাবেজ্-এর পরিকল্পনা অনুযায়ী যন্ত্রটি স্বয়ংক্রিয়ভাবে বীজ গাণিতিক রাশিমালার মান নির্ণয় করতে এবং ছাপতে সক্ষম ছিল। ব্যাবেজ্ এই কাজটি করেছিলেন ব্রিটিশ সরকারের আর্থিক সহায়তায়। যন্ত্রটির পরিকাঠামো রূপায়ণে যথেষ্ট সময় অতিবাহিত হতে দেখে ব্রিটিশ সরকার আর্থিক সহায়তা তুলে নেন। ব্যাবেজ্ কিন্তু তাঁর কাজ চালিয়ে যান। 1833 সালে তিনি তাঁর পরিকল্পিত যন্ত্রটির আরও অনেক পরিমার্জিত রূপ দান করেন এবং এই উন্নতমানের গণকযন্ত্রের নাম দেন অ্যানালিটিকাল্ ইঞ্জিন্ (Analytical Engine)। পান্শ্ কার্ড, স্মৃতি আধার, গণনা কেন্দ্র, ধারাবাহিক নির্দেশ পালন, 20 অঙ্কের শুদ্ধতা, স্বয়ংক্রিয়ভাবে ছাপার ব্যবস্থাপনা ইত্যাদি বৈশিষ্ট্য অ্যানালিটিকাল্ ইঞ্জিনের পরিকল্পনাতে ছিল। ব্যাবেজ্ এর এই গণকযন্ত্রটি বর্তমান বৈদ্যুতিন্ পরিগণকের আদিরূপ। প্রকৃতপক্ষে চার্লস্ ব্যাবেজ্-কে পরিগণক যন্ত্রের জনক বলা হয়। ব্যাবেজ্-এর কল্পিত ইঞ্জিন্ শেষ পর্যন্ত তৈরী হয়নি, কিন্তু এই অ্যানালিটিকাল্ ইঞ্জিন্ এর ধারণাকে ভিত্তি করে পরিগণক যন্ত্রের ক্রমবিকাশ হয়েছে।

চিত্র-২ ঃ চার্লস্ ব্যাবেজ্

পরিগণক যন্ত্রের ক্রমবিবর্তনের সুদীর্ঘ ইতিহাসে ফেল্ট (Felt), হেরম্যান্ হল্এরিথ্ (Herman Hollerith), জেমস্ পাওয়ারস্ (James Powers), জন্ ভন্ ন্যয়ম্যান্ (John Von Neumann) এবং আরও অনেক বিজ্ঞানী, যন্ত্রশিল্পী, প্রযুক্তিবিদ এর উল্লেখযোগ্য অবদান রয়েছে। 1950 সালের আগে পর্যন্ত পরিগণক যন্ত্রের মান উন্নয়ন এবং সংস্করণের কাজ মূলতঃ শিক্ষা-প্রতিষ্ঠানের গবেষণাগারে সীমিত ছিল। 1950 সাল থেকে বিভিন্ন বাণিজ্য সংস্থা ও পরিগণক যন্ত্রের গঠনশৈলীতে রূপান্তর আনতে উদ্যোগী হয়। এই সময় থেকেই বৈদ্যুতিন্ প্রযুক্তিবিদ্যায় একের পর এক যুগান্তকারী আবিষ্কার হতে থাকে। এই সমস্ত আবিষ্কার পরিগণক যন্ত্রের গঠন প্রযুক্তি পাল্টে দিতে থাকে। পরিবর্তনের বিভিন্ন ধাপ অনুযায়ী বিভিন্ন পর্যায়ের পরিগণকের সংক্ষিপ্ত বিবরণ নীচে দেওয়া হল।

প্রথম পর্যায়ের পরিগণক (1945-1958) ঃ

প্রথম পর্যায়ের পরিগণকগুলি ছিল বিপুলায়তন, ওজন প্রায় 30 টনের মত এবং প্রায় 1500 বর্গফুট জায়গা অধিকার করে থাকত। এই ধরণের একটি পরিগণক তৈরী করতে লাগত প্রায় 1800 নির্বাত নল, 1500 রিলে যন্ত্র এবং প্রচুর তার যার মোট দৈর্ঘ্য 800 কিলোমিটার-এর কম নয়। কিন্তু এর কার্যক্ষমতা ছিল অত্যন্ত সীমিত এবং গণনা হত অত্যন্ত ধীর গতিতে। নির্বাত নলগুলি প্রায়ই খারাপ হত এবং মেরামত করতে যথেষ্ট সময় লেগে যেত। প্রথম পর্যায়ের পরিগণক গুলির মধ্যে উল্লেখযোগ্য হল—MARK-I (1945), ENIAC (Electronic Numerical Integrator and Calculator, 1946), UNIVAC (Universal Automatic Computer), EDSAC (Electronic Delay Storage Automatic Calculator, 1949), EDVAC (Electronic Discrete Variable Automatic Computer, 1951), IBM-650 ইত্যাদি। ENIAC যন্ত্রটি সর্বপ্রথম বৈদ্যুতিন্ পরিগণক। এটি সেকেণ্ডে প্রায় 5000 যোগফল নির্ণয় করতে পারত।

দ্বিতীয় পর্যায়ের পরিগণক (1958-1964) ঃ

1950 সালের শেষের দিকে ট্র্যান্সিস্টর আবিষ্কারের পর পরিগণকের আভ্যন্তরীণ গঠনশৈলীতে পরিবর্তন এল। নির্বাত নলের পরিবর্তে ট্র্যান্সিস্টর ব্যবহার করাতে পরিগণক যন্ত্রের আয়তন কিছুটা কমল, তাপ উৎপাদন হ্রাস পেল এবং যন্ত্র কিছুটা বিশ্বস্ত হল। প্রথম পর্যায়ের তুলনায় দ্বিতীয় পর্যায়ের পরিগণকের গণনার হার অনেকটা বৃদ্ধি পেয়েছিল এবং যন্ত্র চালনা করতে তুলনামূলক ভাবে কম শক্তির প্রয়োজন হত। ENIAC, IBM 7090, IBM 704 ইত্যাদি হল দ্বিতীয় পর্যায়ের পরিগণক। দ্বিতীয় পর্যায়ের ENIAC যন্ত্রটি প্রতি সেকেণ্ডে প্রায় এক মিলিয়ন যোগফল নির্ণয় করতে পারত।

তৃতীয় পর্যায়ের পরিগণক (1964-1975) ঃ

1960 সালে ইন্টিগ্রেটেড সারকিট্ (Integrated Circuits) বা IC চিপস্-এর আবিষ্কার এক যুগান্তকারী ঘটনা। একটি IC চিপ হল ছোট্ট একটি সিলিকণের পাতলা টুকরো যাতে কয়েক হাজার ট্র্যানসিস্টর বৈদ্যুতিক সহযোগ দ্বারা বয়ন করা থাকে। এইরকম একটি IC চিপ মাত্র 0.25 বর্গ সেন্টিমিটার ক্ষেত্রফল বিশিষ্ট ছিল। এই ধরণের IC চিপস্ ব্যবহার করে যে সমস্ত পরিগণক যন্ত্র তৈরী হয়েছিল সেগুলি হল তৃতীয় পর্যায়ের পরিগণক। IC চিপস্ ব্যবহার করার ফলে তৃতীয় পর্যায়ের পরিগণক যন্ত্রের আয়তন যথেষ্ট কমে গেলে, কার্যক্ষমতা বেড়ে গেল অনেক গুণ এবং যন্ত্র আরও অনেক বেশী বিশ্বস্ত হল। ক্রমবিবর্তনের তৃতীয় পর্যায়ে পরিগণক যন্ত্রের আনুষঙ্গিক স্মৃতি আধার অংশের যথেষ্ট উন্নতি হয়েছিল এবং বহুমুখী সংসাধন এবং বহুমুখী ক্রিয়া সম্পাদনের সুবিধাও এতে ছিল।

তৃতীয় পর্যায়ের পরিগণক যন্ত্রেই প্রথম পরিচালক সিস্টেম বা অপারেটিং সিস্টেম (Operating System) এর ধারণা প্রয়োগ করা হয়েছিল। IBM-360 শ্রেণী, ICL 1900 ইত্যাদি পরিগণক যন্ত্রগুলি প্রতি সেকেণ্ডে কয়েক হাজার মিলিয়ন যোগফল নির্ণয় করার ক্ষমতা সম্পন্ন ছিল।

চতুর্থ পর্যায়ের পরিগণক (1975 থেকে বর্তমান) ঃ

বৈদ্যুতিন্ প্রযুক্তিবিদ্যার উন্নতির সাথে সাথে IC চিপসের প্রভূত উন্নতি হয়েছে। উন্নতমানের IC চিপ্ কে বলা হয় LSIC (Large-Scale-Integrated-Chip) এবং আরও উন্নতমানের IC চিপ-কে বলা হয় VLSIC (Very-Large-Scale-Integrated-Chip)। এই রকম একটি VLSIC-তে কয়েক শত হাজার বৈদ্যুতিন্ অংশবিশেষ পুঞ্জীভূত অবস্থায় থাকতে পারে। এই ধরণের উন্নতমানের IC চিপস্ ব্যবহার করে চতুর্থ পর্যায়ের পরিগণক যন্ত্র তৈরী করা হয়। বর্তমানে আমরা চতুর্থ পর্যায়ের পরিগণক যন্ত্র ব্যবহার করছি। উন্নতমানের IC চিপস্ ব্যবহার করার ফলে পরিগণক যন্ত্রের স্মৃতিশক্তি অনেক বৃদ্ধি পেয়েছে, যন্ত্রের কেন্দ্রীয় সংসাধক অংশটির আয়তন অনেক হ্রাস পেয়েছে, তথ্য সংসাধন এবং গণনার হার অনেক বৃদ্ধি পেয়েছে। বর্তমানে একটি মাত্র উন্নতমানের IC চিপের সাহায্যেই পরিগণক যন্ত্রের সমস্ত রকমের সংসাধনের কাজ হতে পারে। এই ধরণের একটি চিপ-কে বলা হয় মাইক্রো-সংসাধক বা মাইক্রো-প্রসেসর। Intel Corporation সর্বপ্রথম মাইক্রো-চিপ (Microprocessor on a Chip) নিয়ে আসে বাজারে। মাইক্রো চিপ পরিগণক যন্ত্রের কাজে যে গতি এনে দিয়েছে তা মাপতে সময়ের একক ধরতে হচ্ছে ন্যানো সেকেণ্ড (= 10^{-9} সেকেণ্ড) কিংবা কখনও পিকো সেকেণ্ড (= 10^{-12} সেকেণ্ড)।

পঞ্চম পর্যায়ের পরিগণক (1985 থেকে শুরু) ঃ

চতুর্থ পর্যায়ের পরিগণক যন্ত্রগুলি গণনা এবং আরও বহুবিধ কর্মক্ষেত্রে অসামান্য সূক্ষ্মতা, শুদ্ধতা এবং দ্রুততার দ্বারা মানুষের অত্যন্ত আস্থাশীল ও অপরিহার্য যন্ত্র হিসাবে স্থান করে নিয়েছে। কিন্তু এই পরিগণক যন্ত্রগুলির নিজস্ব বোধশক্তি বলে কিছু নেই। এরা শুধুমাত্র ব্যবহারকারীর নির্দেশানুসারে কাজ করতে পারে। পরিগণক যন্ত্রের মধ্যে যাতে কিছুটা কৃত্রিম বোধশক্তি থাকে সেই প্রচেষ্টা বিভিন্ন দেশের বিজ্ঞানী ও প্রযুক্তিবিদ-এর মধ্যে চলছে প্রায় দু'দশকেরও বেশী সময় ধরে। কিছুটা সাফল্য ইতিমধ্যে পাওয়া গেছে। এই ধরণের উন্নতমানের পরিগণক যন্ত্র পঞ্চম পর্যায়ের পরিগণক যন্ত্র রূপে অভিহিত হবে। এখনো পর্যন্ত পঞ্চম পর্যায়ের পরিগণক যন্ত্রের বাণিজ্যিক উৎপাদন শুরু হয়নি।

পরিগণকের প্রকারভেদ ঃ

পরিগণকের ভৌত গঠন, গঠনশৈলী এবং ব্যাবহারিক প্রয়োগের বিভিন্নতার উপর ভিত্তি করে পরিগণকের শ্রেণীবিভাজন করা হয়।

ব্যবহারিক প্রয়োগক্ষেত্র অনুযায়ী পরিগণক তিন শ্রেণীর হতে পারে—

(ক) পরিগণক অনুরূপ বা অ্যানালগ কম্পিউটার (Analog Computer)

(খ) সংখ্যাত্মক পরিগণক বা ডিজিটাল্ কম্পিউটার (Digital Computer)

(গ) মিশ্র প্রকৃতির পরিগণক বা হাইব্রিড কম্পিউটার (Hybrid Computer)

এই তিন শ্রেণীর পরিগণকের কিছু উদাহরণ নীচে দেওয়া হল।

(ক) পরিগণক অনুরূপ যন্ত্র ঃ

বিশেষ কোন উদ্দেশ্য সাধনের জন্য নির্মিত পরিগণক অনুরূপ নামক যন্ত্র শুধুমাত্র সন্তত তথ্যের উপর প্রক্রিয়াশীল। বিভিন্ন ধরণের পরিগণক অনুরূপ যন্ত্র আমরা বিভিন্ন কাজে ব্যবহার করে থাকি। যেমন— মানবদেহের ওজন নির্ণয়ক যন্ত্র, তাপমাপক যন্ত্র বা থার্মোমিটার, গাড়ির বেগমাপক যন্ত্র বা স্পীডমিটার, পেট্রোল পাম্পে ব্যবহৃত পেট্রোল, ডিজেলের আয়তন এবং মূল্য নির্ণয়ক যন্ত্র ইত্যাদি।

(খ) সংখ্যাত্মক পরিগণক ঃ

এই ধরণের পরিগণক বিশ্লিষ্ট তথ্য আদান-প্রদান, প্রক্রিয়াকরণ বা সংসাধনে সক্ষম। পরিগণক অনুরূপ যন্ত্রের তুলনায় সংখ্যাত্মক পরিগণকের কাজ বহুমুখী এবং কাজের গতি ও সূক্ষ্মতাও অনেক বেশী। সংখ্যাত্মক পরিগণক দু'ধরণের।

(i) সর্বজনীন কার্য সহায়ক সংখ্যাত্মক পরিগণক

(ii) বিশেষ কার্য সহায়ক সংখ্যাত্মক পরিগণক

এই দু'ধরণের সংখ্যাত্মক পরিগণকের উদাহরণ নীচে দেওয়া হল।

(i) সর্বজনীন কার্য সহায়ক সংখ্যাত্মক পরিগণক ঃ—

এই ধরণের পরিগণকের সাহায্যে একাধিক ধরণের কাজ করা যেতে পারে। কেউ এর সাহায্যে লিপিমুদ্রণের কাজ করতে পারে; কেউ একে অফিস-আদালতের কাজে লাগাতে পারে; ব্যাঙ্কে একে হিসাব রক্ষণকার্যে ব্যবহার করা যেতে পারে; আবার গবেষণাক্ষেত্রে এর সাহায্যে জটিল গাণিতিক সমস্যার সমাধান করা যেতে পারে; কেউ বা পারেন ছবি আঁকতে অথবা দূরেক্ষণ খেলায় মেতে উঠতে। এই সমস্ত বিভিন্ন ধরণের কাজ করা যেতে পারে শুধুমাত্র এই ধরণের একটি সংখ্যাত্মক পরিগণকের সাহায্যেই। অর্থাৎ, পরিগণক যন্ত্র একটিই শুধু বিভিন্ন ধরনের কাজের জন্য চাই বিভিন্ন প্রোগ্রাম বা ক্রিয়া নির্দেশাবলী। বর্তমানে এই ধরণের একটি পরিগণক যন্ত্র একই সাথে একাধিক প্রোগ্রাম সঞ্চয় করে রাখতে পারে তার স্মৃতি কক্ষে। ব্যবহৃত প্রোগ্রামের প্রকৃতির উপর নির্ভর করে এই ধরণের একটি পরিগণক হয়ে উঠতে পারে শিক্ষা মাধ্যম বা বিনোদন মাধ্যম কিংবা শিল্প বাণিজ্য সহায়ক বা অফিসভৃত্য।

(ii) বিশেষ কার্যসহায়ক সংখ্যাত্মক পরিগণক ঃ—

এই ধরণের পরিগণক বিশেষ কোন কার্যসিদ্ধির উদ্দেশ্য নিয়ে নির্মিত হয়ে থাকে। কাযসিদ্ধির উদ্দেশ্য নিয়ে নির্মিত হয়ে থাকে। কার্যসিদ্ধির জন্য প্রয়োজনীয় ক্রিয়ানির্দেশাবলী বা প্রোগ্রাম যন্ত্রটি উৎপাদনের সময়েই উৎপাদক সংস্থাকর্তৃক অন্তর্নিহিত করা থাকে। যেমন, রেল বা বিমানের টিকিট সংরক্ষণ কেন্দ্রে, স্বাস্থ্যকেন্দ্রে রুগীর রক্তচাপ নির্ণয় করতে অথবা মহাকাশযান উৎক্ষেপন ও তার গতিপথ নির্ণয়, বিমান চালনায় ও কৃত্রিম উপগ্রহ পরিচালনায় যে সমস্ত পরিগণক ব্যবহার করা হয় সেগুলি এই শ্রেণীর। এই ধরণের কোন পরিগণক যে কাজের জন্য নির্মিত সেটি শুধু ঐ বিশেষ কাজটিই করতে সক্ষম।

(গ) মিশ্র প্রকৃতির পরিগণক ঃ

এই ধরণের পরিগণকের একাংশ পরিগণক অনুরূপ যন্ত্রের মত কাজ করে এবং অপর অংশ সংখ্যাত্মক পরিগণকের মত কাজ করে। যেমন—হাসপাতাল বা নার্সিং হোমে অসুস্থ রূগীর দেহের তাপমান, রক্তচাপ, হৃৎস্পন্দন ইত্যাদি সামগ্রিকভাবে পর্যবেক্ষণ করতে যে পরিগণক ব্যবহার করা হয় তা হল মিশ্র প্রকৃতির পরিগণক। এর প্রাথমিক কাজ হল তাপমান, রক্তচাপ, হৃৎস্পন্দন ইত্যাদি পরিমাপ করা। কার্যপ্রণালীর এই অংশটুকু পরিগণক অনুরূপ যন্ত্রের ন্যায়। পরবর্তী ধাপে এই সমস্ত পরিমাপ বিশ্লিষ্ট করে পরিগণকের অপর অংশে পাঠানো হয় যে অংশের কার্যপ্রণালী সংখ্যাত্মক পরিগণকের ন্যায়। এই অংশে যে বিশ্লিষ্ট তথ্য গৃহীত হল তা শেষ পর্যন্ত লেখচিত্ররূপে দৃশ্যমান হয়।

পরিগণকের ভৌত গঠন এবং গঠনশৈলীর তারতম্য অনুযায়ী পরিগণক চার শ্রেণীর হতে পারে—

(ক) মাইক্রো পরিগণক বা মাইক্রোকম্পিউটার

(খ) মিনি পরিগণক বা মিনি কম্পিউটার

(গ) মেইনফ্রেম পরিগণক বা মেইনফ্রেম কম্পিউটার

(ঘ) সুপার পরিগণক বা সুপার কম্পিউটার

নীচে এই চার ধরনের পরিগণক সম্পর্কে সংক্ষিপ্ত বিবরণ দেওয়া হল।

(ক) মাইক্রো পরিগণক ঃ বিভিন্ন ধরণের বৈদ্যুতিন্ পরিগণকের মধ্যে সবচেয়ে কম কর্মদক্ষতা হল মাইক্রো পরিগণকের। একটি মাইক্রো পরিগণকের স্মৃতি আধারে সঞ্চয়সীমা 64 - 128 কিলোবাইট এবং শব্দ সঞ্চয় সীমা 512W - 8 KW। এই ধরণের একটি পরিগণকে তথ্য আদান-প্রদানের হার সর্বাধিক 5 লক্ষ বাইট/সেকেণ্ড হতে পারে। একটি মাইক্রো পরিগণকের বাহ্যিক গঠন হতে পারে একটি ডেস্কটপ্ অথবা একটি ল্যাপটপের মত। এটি হতে পারে সর্বজনীন কার্য সহায়ক বা বিশেষ কার্য সহায়ক। দৈনন্দিন কাজে কর্মে ব্যবহৃত পরিগণক গুলি আসলে মাইক্রো পরিগণক। এই যন্ত্রগুলিকে ব্যবহারকারীরা তাদের নিজেদের ব্যক্তিগত যন্ত্র হিসাবে আপন করে নিয়েছে। এদেরকে সংক্ষেপে বলা হয় PC অর্থাৎ Personal Computer।

প্রথম মাইক্রো পরিগণক Intel 8080 তৈরী হয়েছিল 1975 সালে। অবশ্য বাণিজ্যিক ভাবে প্রস্তুত প্রথম মাইক্রো পরিগণক হল ALTAIR 8080। পরবর্তীকালে APPLE, MACINTOSH, IBM, HP ইত্যাদি আরও অনেক সংস্থা মাইক্রো পরিগণক-এর উৎপাদন শুরু করে বাণিজ্যিক ভাবে। 1980 সালে IBM সংস্থাই প্রথম মাইক্রো পরিগণক যন্ত্রকে Personal Computer নামে অভিহিত করে।

দ্রষ্টব্য ঃ এখানে একটি বিট বলতে '0' অথবা '1' সংখ্যাকে বলা হয়েছে। একটি বাইট বলতে আটটি বিটের সমাহার ধরা হয়েছে।

1 KB = 1 কিলোবাইট = 2^{10} বাইট = 1024 বাইট

1 MB = 1 মেগাবাইট = 2^{10} কিলোবাইট

একটি পরিগণকীয় শব্দদৈর্ঘ্য হতে পারে 8-64 বিট।

1 KW = 1 কিলোওয়ার্ড = 1000 ওয়ার্ড

= 1000 পরিগণকীয় শব্দ

1 MW = 1000 KW

('বিট' ও 'বাইট' সম্বন্ধে বিস্তৃত আলোচনা পরবর্তী অংশে করা হয়েছে)

(খ) মিনি পরিগণক ঃ মিনি পরিগণকের কর্মদক্ষতা এবং সঞ্চয়সীমা মাইক্রো পরিগণকের তুলনায় কিছুটা বেশী। একটি মিনি পরিগণকের স্মৃতি আধারে সঞ্চয়সীমা 256-512 কিলোবাইট এবং পরিগণকীয় শব্দ সঞ্চয়সীমা 4 KW - 2 MW। একটি মিনি পরিগণকে তথ্য আদান প্রদানের হার সর্বাধিক 4 মিলিয়ন বাইট/সেকেণ্ড হতে পারে। একটি মাইক্রো পরিগণক এবং একটি মিনি পরিগণকের মূল পার্থক্য হল এই যে, একটি মিনি পরিগণকে একই সাথে একাধিক ব্যক্তি কাজ করতে পারেন মিনি পরিগণকটির সাথে সংযুক্ত একাধিক প্রান্তীয় যন্ত্রাংশ বিশেষ-এর সাহায্যে।

PDP-8 নামে প্রথম মিনি পরিগণকটি তৈরী হয়েছিল 1965 সালে। পরবর্তীকালে আরও উচ্চ ক্ষমতাসম্পন্ন মিনি পরিগণক তৈরী হয়েছে। যেমন—VAX7500, HP 2000, IBM 17, HP 3000 ইত্যাদি।

(গ) মেইনফ্রেম পরিগণক ঃ মাইক্রো পরিগণক বা মিনি পরিগণকের তুলনায় মেইনফ্রেম পরিগণকের কর্মদক্ষতা এবং সঞ্চয়সীমা অনেক বেশী। একটি মেইনফ্রেম পরিগণকের স্মৃতি আধারে সঞ্চয়সীমা 8.4 মিলিয়ন বাইট এবং পরিগণকীয় শব্দ সঞ্চয়সীমা 500 KW - 10 MW। একটি মেইনফ্রেম পরিগণকের তথ্য আদান-প্রদানের হার সর্বাধিক 16 মিলিয়ন বাইট/সেকেণ্ড হতে পারে। একটি মেইনফ্রেম পরিগণকের সাহায্যে বহুসংখ্যক প্রান্তীয় ব্যবহারকারী একই সাথে কাজ করতে পারেন। অর্থাৎ, মেইনফ্রেম পরিগণকের ভূমিকা হল কেন্দ্রীয় মূল পরিসেবক যন্ত্রের মত। মেইনফ্রেম পরিগণক যন্ত্রের ব্যবহার হয় সাধারণত ঃ মাঝারি অথবা বড় শিল্প-প্রতিষ্ঠান, শিক্ষা প্রতিষ্ঠান বা বাণিজ্যিক সংস্থায়। IBM 4381, ICL 39-শ্রেণী, UNIVAC 1108/1110 ইত্যাদি হল বিভিন্ন সংস্থার তৈরী মেইনফ্রেম পরিগণক।

(ঘ) সুপার পরিগণক ঃ একটি সুপার পরিগণকের কর্মদক্ষতা, স্মৃতি আধারের সঞ্চয়সীমা বা তথ্য সংসাধনের হার উপরিল্লিখিত প্রথম তিন শ্রেণীর যে কোন পরিগণকের তুলনায় দশ গুণ বা তারও বেশী। এই ধরণের পরিগণক ব্যবহার করা হয় পদার্থবিদ্যা, রসায়নশাস্ত্র, চিকিৎসাশাস্ত্র, মহাকাশ গবেষণা, আবহাওয়ার পূর্বাভাস সংক্রান্ত গবেষণা ইত্যাদি জটিল গবেষণা বা অনুরূপ কোন গুরুত্বপূর্ণ কাজে। অবশ্য প্রথম দিকে সুপার পরিগণকের ব্যবহার সরকারী কাজে সীমাবদ্ধ ছিল।

প্রথম সুপার পরিগণকটি তৈরী করেছিল আমেরিকার Cray Research Inc. 1976 সালে। যন্ত্রটির নাম ছিল CRAY-1। পরবর্তীকালে 1982 সালে CRAY X-MP নামক সুপার পরিগণকটিও এদের তৈরী। এই যন্ত্রটি সেকেণ্ডে 1 বিলিয়ন সংখ্যক নির্দেশ পালন করার ক্ষমতাসম্পন্ন ছিল। CRAY-শ্রেণী ছাড়া CDC-cyber শ্রেণী, ETA সিস্টেম ইত্যাদি হল বিভিন্ন সুপার পরিগণকের উদাহরণ।

বর্তমান পরিগণক যন্ত্রের বিবরণ ঃ

হার্ডওয়্যার, সফট্ওয়্যার ও হার্টওয়্যার ঃ

যে তিনটি অংশের সমন্বয়ে পরিগণক যন্ত্র সক্রিয় হতে পারে সেই অংশ তিনটি হল—হার্ডওয়্যার, সফট্ওয়্যার ও হার্টওয়্যার।

হার্ডওয়্যার অংশ বলতে পরিগণকের যান্ত্রিক গঠনতন্ত্রকে বোঝান হয়। পরিগণকের যান্ত্রিক অংশ যেমন—মনিটর, কীবোর্ড, মাউস্, প্রিন্টার, প্রসেসিং ইউনিট ইত্যাদি হার্ডওয়্যার অংশের অন্তর্ভুক্ত।

সফট্ওয়্যার বলতে পরিগণক যন্ত্রকে সক্রিয় করার জন্য প্রয়োজনীয় প্রোগ্রাম্ অংশকে বোঝান হয়। কোনও বিশেষ কার্য সমাধানের জন্য প্রয়োজনীয় ক্রিয়া নির্দেশগুলি কোনও সুনির্দিষ্ট ভাষার ব্যাকরণ অণুসারে লেখা হলে সেই ক্রিয়ানির্দেশ তালিকাকে বলা হয় ঐ বিশেষ কার্য সমাধানের জন্য প্রয়োজনীয় প্রোগ্রাম।

একটি দূরদর্শনের যন্ত্রাংশকে যদি পরিগণক যন্ত্রের হার্ডওয়্যার-এর সাথে তুলনা করা হয় তাহলে দূরদর্শনে সম্প্রচারিত বিভিন্ন অনুষ্ঠান বা প্রোগ্রাম অংশ হল পরিগণকের সফট্ওয়্যার-এর সাথে তুলনীয়।

একটি পরিগণক যন্ত্রকে সক্রিয় করে তোলার পিছনে যে সমস্ত বিশেষজ্ঞ ব্যক্তির ভূমিকা রয়েছে তাদের বলা হয় হার্টওয়্যার। হার্ডওয়্যার অংশ তৈরীর সাথে যে সমস্ত যন্ত্রশিল্পী যুক্ত আছেন (হার্ডওয়্যার ইঞ্জিনীয়ার), সফট্ওয়্যার অংশ লিখছেন যারা (সফট্ওয়ার ইঞ্জিনীয়ার) এবং যে সমস্ত পদস্থ আধিকারিক পরিগণক ব্যবহারকারীদের কাজকর্মের সর্বাঙ্গীন তদারকি করেন তারা সকলেই হার্টওয়্যার অংশে আছেন।

একটি দূরদর্শনকে যদি একটি পরিগণকের সাথে তুলনা করা হয় তাহলে দূরদর্শন নির্মানের সাথে যুক্ত যন্ত্রশিল্পীরা, দূরদর্শনের বিভিন্ন প্রোগ্রাম তৈরী এবং সম্প্রচারের নেপথ্যে যে সমস্ত শিল্পী ও কর্মীরা যুক্ত আছেন তারা হলেন দূরদর্শনের হার্টওয়্যার।

হার্ডওয়্যার-এর বিভিন্ন অংশ ঃ

পরিগণকের হার্ডওয়্যার-এর অংশগুলিকে নিম্নলিখিত দুটি প্রধান ভাগে বিভক্ত করা হয়—

(1) কেন্দ্রীয় সংসাধক অংশ

(2) প্রান্তীয় সহায়ক অংশ

(1) কেন্দ্রীয় সংসাধক অংশ ঃ কেন্দ্রীয় সংসাধক অংশটিকে ইংরাজীতে বলা হয় Central Processing Unit বা সংক্ষেপে CPU। পরিগণকের যাবতীয় কার্যসম্পাদন এবং নিয়ন্ত্রন হয় কেন্দ্রীয় সংসাধক অংশে। এই অংশে আছে—

(i) মূখ্য বা কেন্দ্রীয় স্মৃতি কক্ষ বা স্মৃতি আধার

(ii) গণিত ও যুক্তি ক্ষেত্র বা গণিত ও তর্কক্ষেত্র

(iii) নিয়ন্ত্রণ কেন্দ্র

পরিগণকের CPU অংশটিকে মানবদেহের হাত-পা ব্যতীত দেহাংশের (অর্থাৎ মস্তিষ্ক + মূল দেহ) সাথে তুলনা করা যেতে পারে।

(2) প্রান্তীয় সহায়ক অংশ ঃ পরিগণক যন্ত্রের মূল অংশ অর্থাৎ CPU অংশের সাথে যুক্ত যে বিভিন্ন যন্ত্রাংশ CPU এর কাজে সহায়তা করে সেগুলি হল প্রান্তীয় সহায়ক অংশ। এই অংশে আছে

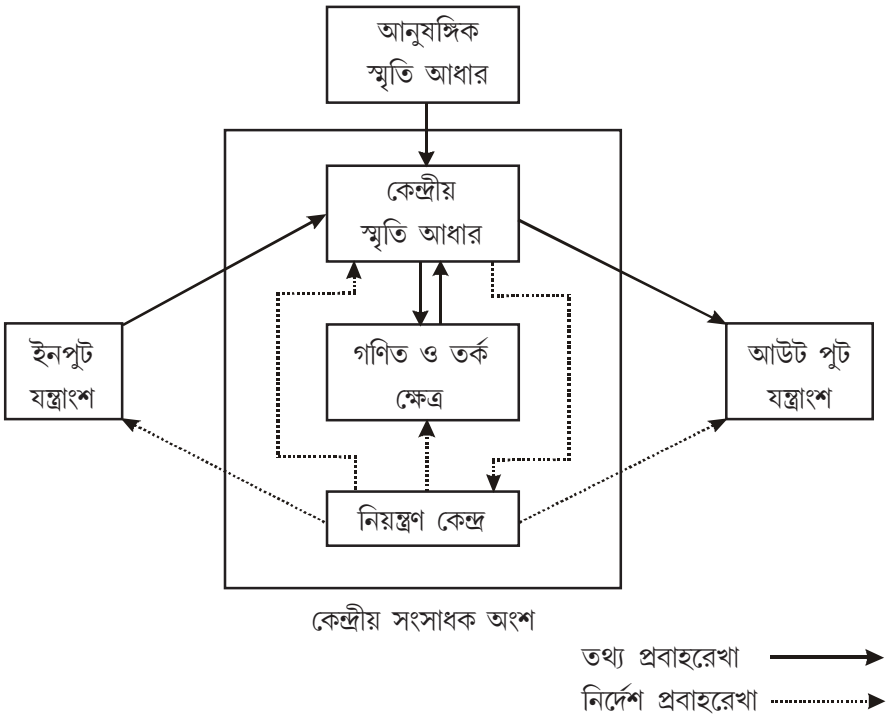
(i) প্রেরক অংশ বা সরবরাহকারী অংশ বা ইনপুট যন্ত্রাংশ

(ii) ফলাফল জ্ঞাপনকারী অংশ বা আউটপুট যন্ত্রাংশ

(iii) আনুষঙ্গিক স্মৃতি আধার

যদি মানবদেহকে পরিগণকের সাথে তুলনা করা হয় তাহলে হাত, পা, চোখ, কান, জিহ্বা ইত্যাদি হল মানবদেহের প্রান্তীয় সহায়ক অংশ।

পরিগণক যন্ত্রে কার্য চলাকালীন তার হার্ডওয়্যার-এর বিভিন্ন অংশে তথ্য এবং নির্দেশের আদান প্রদান কিভাবে হয় তা নীচের ব্লকচিত্রে দেখান হয়েছে।



চিত্র : হার্ডওয়্যার-এর বিভিন্ন অংশে তথ্য ও নির্দেশ আদান প্রদানের ব্লকচিত্র

পরিগণক যন্ত্রের বিভিন্ন অংশের কার্যবিধি ঃ

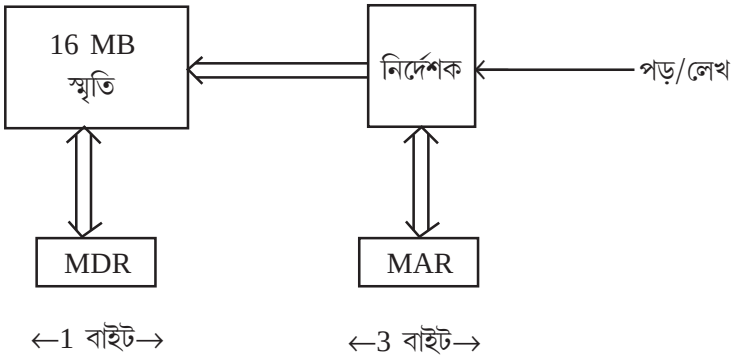
মূখ্য/কেন্দ্রীয় স্মৃতি আধার ঃ

এটি থাকে পরিগণক যন্ত্রের কেন্দ্রীয় সংসাধক অংশের অভ্যন্তরে। কেন্দ্রীয় স্মৃতি আধারে থাকে প্রচুর সংখ্যক স্মৃতিকোষ। প্রতিটি কোষ সঞ্চয় করতে পারে একটি করে দ্বি প্রতীকী সংখ্যা। স্মৃতি আধারের স্মৃতিকোষ দু'ধরণের হতে পারে—উদ্বায়ী এবং অনুদ্বায়ী। যে সমস্ত স্মৃতিকোষে বিদ্যুৎ প্রবাহ বন্ধ হওয়ার সাথে সাথে সঞ্চিত চিহ্নগুলি মুছে যায় সেই সমস্ত স্মৃতিকোষকে বলা হয় উদ্বায়ী স্মৃতিকোষ। অপরদিকে যে সমস্ত স্মৃতিকোষ অবিরাম বিদ্যুৎ সরবরাহ ছাড়াও তথ্য সঞ্চয় করে রাখতে পারে তাদের বলা হয় অনুদ্বায়ী স্মৃতিকোষ।

পরিগণকের স্মৃতি আধারে সঞ্চিত বিভিন্ন পরিগণকীয় শব্দের (এক একটি পরিগণকীয় শব্দ কয়েকটি দ্বিপ্রতীকী সংখ্যার সমন্বয়ে তৈরী) সুনির্দিষ্ট ঠিকানা থাকে। কোন পরিগণকীয় শব্দকে স্মৃতি আধারে সুনির্দিষ্ট ঠিকানায় সঞ্চয়

করতে হলে সুনির্দিষ্ট ঠিকানা সংখ্যাটি রাখতে হয় ঠিকানা নিবন্ধক স্মৃতিতে (Memory Address Register, সংক্ষেপে MAR) এবং ঐ পরিগণকীয় শব্দটিকে রাখতে হয় তথ্য নিবন্ধক স্মৃতিতে (Memory Data Register, সংক্ষেপে MDR)। এরপর একটি লেখার নির্দেশ দেওয়া হলে শব্দটি সুনির্দিষ্ট ঠিকানায় সঞ্চিত হতে পারে। অনুরূপভাবে, কোন বিশেষ ঠিকানায় সঞ্চিত শব্দটি পাঠ করতে হলে ঐ বিশেষ ঠিকানা-সংখ্যাটি ঠিকানা নিবন্ধক স্মৃতিতে সরবরাহ করতে হয়। এরপর একটি পাঠ নির্দেশ পাঠানো হলে ঐ বিশেষ ঠিকানায় সঞ্চিত শব্দটি তথ্য নিবন্ধক স্মৃতিতে চলে আসে।

কোন পরিগণকের স্মৃতি আধারে শব্দ সঞ্চয়সীমা 16 MB বা 16 মেগাবাইট হলে স্মৃতি আধার, MAR ও MDR এর মধ্যে কিভাবে সংযোগসাধন হয় তা নীচের ব্লকচিত্রে দেখানো হল।

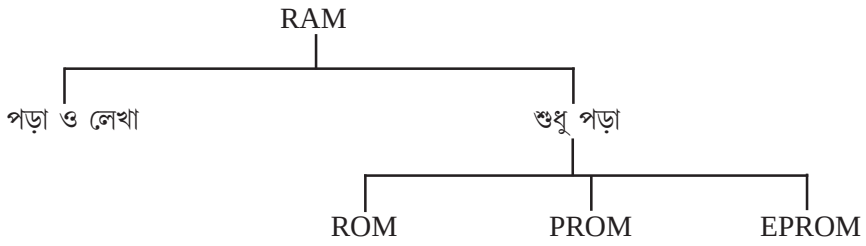


চিত্র : 16 MB স্মৃতি আধারের ব্লকচিত্র

যেহেতু 16 MB = 2^4 MB = $2^4 \times 2^{20}$ বাইট সেহেতু 24 অঙ্কের দ্বিপ্রতীকী সংখ্যা বা 3 বাইট (1 বাইট = 8 টি বিট) মাপের MAR দ্বারা স্মৃতি আধারের সমস্ত ঠিকানা সংখ্যা সূচিত করা যাবে। এই উদাহরণটিতে, একটি পরিগণকীয় শব্দের দৈর্ঘ্য 1 বাইট বা 8 বিট ধরা হয়েছে।

স্মৃতি আধার থেকে কোন শব্দকে পড়তে বা লিখতে যে সময় লাগে তা যদি শব্দের ঠিকানা সংখ্যার উপর নির্ভরশীল না হয় তাহলে এইরূপ স্মৃতি আধারকে বলা হয় RAM বা ইংরাজীতে Random Access Memory পরিগণকের কেন্দ্রীয় স্মৃতি আধারে সমস্ত তথ্য এবং প্রোগ্রাম সঞ্চিত রাখে RAM যা তৈরী হয় বৈদ্যুতিন্ অর্ধপরিবাহী ফ্লিপ্-ফ্লপ্ এর সাহায্যে।

কোন RAM এর অন্তর্নিহিত তথ্য যদি শুধু মাত্র পড়া যায় এবং মোছা না যায় তাহলে তাকে বলে ROM বা ইংরাজীতে Read Only Memory। কোন ROM-এ যদি বিশেষ তড়িৎ বর্তনী ব্যবহার করে লেখার সুযোগ থাকে তবে তাকে PMON বলা হয়। PROM (Programmable ROM)-ঐ এককর কিছু লেখা হলে তা আর মোছা যায় না। যে সমস্ত PROM এর লেখা অতিবেগুনী রশ্মির সাহায্যে মোছা যায় এবং পুনরায় লেখা যায় তাদের বলা হয় EPROM (Erasable এবং Programmable ROM)।



যুক্তি ও গণনা কেন্দ্র (Arithmetic Logic Unit বা ALU) ঃ

পরিগণক যন্ত্রের এই কেন্দ্রটিতে দু'ধরণের কার্য সম্পাদন হয়। প্রথমতঃ পরিগণকের যাবতীয় গাণিতিক হিসাব, যেমন, যোগ, বিয়োগ, গুণ, ভাগ ইত্যাদি সম্পাদিত হয় ALU-তে। এই সমস্ত কার্য সম্পাদনের জন্য ALU এর অভ্যন্তরে বেশ কিছু ছোট ছোট কার্য একক রয়েছে। যেমন—যোগ একক, গুণ একক, ভাগ একক, সূচকযুক্ত গাণিতিক পদ্ধতি একক ইত্যাদি।

দ্বিতীয়তঃ পরিগণক ব্যবহারকারীর প্রোগ্রামের জন্য প্রয়োজনীয় যুক্তি সম্মত সিদ্ধান্ত নির্ণয় এবং তুলনামূলক বিচারকার্য হয় ALU-তে। ALU-এর অভ্যন্তরে যে যুক্তি এককটি থাকে সেটিতে Boolean তর্কবিদ্যা সংক্রান্ত কার্যাদি সম্পাদিত হয়। এছাড়া একটি লুপ্ নিয়ন্ত্রক একক প্রোগ্রামের জন্য প্রয়োজনীয় তুলনামূলক বিচারকার্য করে।

একটি ছোট উদাহরণের সাহায্যে ALU এর কার্য প্রকৃতি বোঝা যেতে পারে। ধরা যাক, $x = 15.4$, $y = 2.61$ এবং $z = -5.3$। আরও ধরা যাক, A ও B-দুটি বিবৃতি হল—

$$A : (x - y) > (xy + y)$$

$$B : (x + z) > xy$$

ALU-এককটি এই বিবৃতিদুটি নিরীক্ষণের পর জানিয়ে দেবে— A হল মিথ্যা এবং B হল সত্যি।

নিয়ন্ত্রণ কেন্দ্র (Control Unit বা CU) ঃ

পরিগণকের কেন্দ্রীয় সাংসাধক অংশের যাবতীয় কার্য নিয়ন্ত্রণ করে এই নিয়ন্ত্রণ কেন্দ্র নামক এককটি। প্রোগ্রাম রচয়িতা যে সমস্ত তথ্য সরবরাহ করেন সেগুলি স্মৃতি আধারে সঞ্চিত হয় নিয়ন্ত্রণ কেন্দ্রের নির্দেশে। প্রোগ্রাম-এর মধ্যে উল্লিখিত কার্যনির্দেশাদির অর্থ উদ্ধার করা এবং সেই সমস্ত নির্দেশ অনুসারে কার্যকলাপ নিয়ন্ত্রণ করা হয় এই এককটিতে।

তথ্য গ্রহণ একক ঃ

প্রোগ্রাম রচয়িতা যে সমস্ত তথ্য এবং নির্দেশ সরবরাহ করেন তা গৃহীত হয় তথ্য গ্রহণ কেন্দ্রের মাধ্যমে। বিভিন্ন ধরণের যন্ত্র বা যন্ত্রাংশ তথ্য গ্রহণের কাজ করতে পারে। বর্তমানে সবচেয়ে পরিচিত তথ্য গ্রাহক হল কীবোর্ড এবং মাউস। কীবোর্ডে প্রায় 104 টি বিভিন্ন ধরণের কী বা বোতাম থাকে; যেমন—বর্ণমালা কী, সংখ্যা কী, নিয়ন্ত্রণ কী ইত্যাদি। পরিগণক যন্ত্রটি যখন চালু থাকে অর্থাৎ যন্ত্রটিতে যখন তড়িৎসংযোগ থাকে তখন কীবোর্ডের কোন একটি

বোতাম টিপলে সংশ্লিষ্ট বর্ণ বা সংখ্যা বা বিশেষ চিহ্নটি পরিগণক যন্ত্রের মনিটর বা পর্দায় দৃশ্যমান হয়। মনিটরে স্থান থেকে স্থানান্তরে দ্রুত সঞ্চরণ এবং চিত্রাঙ্কণে বিশেষ সহযোগিতা করে মাউস।

কীবোর্ড এবং মাউস ব্যতীত অন্যান্য তথ্যগ্রাহক যন্ত্রগুলি হল ফ্লপি ডিস্ক, চুম্বকীয় ফিতা, পানশ্ কার্ড, বার সঙ্কেত, OCR (অর্থাৎ Optical Character Recognition), OMR (Optical Mark Reading and recognition), MICR (Magnetic Ink Character Recognition) ইত্যাদি।

পরিগণক যন্ত্রের ইতিহাসে একেবারে গোড়ার দিকে পানশ্ কার্ডই ছিল মূল তথ্য গ্রাহক। এটি ছিল 18.8 সেমি × 8.3 সেমি মাপের আয়তাকার পাতলা কার্ড যাতে 80 টি স্তম্ভ এবং 12টি শ্রেণীতে ছিদ্র থাকতে পারে। পরিগণক যন্ত্রে কোন তথ্য সরবরাহ করতে হলে প্রথমে সেই তথ্যকে পানশ্ কার্ডে যথাযথ স্থানে ছিদ্র করে কার্ডগুলি তৈরী করতে হত। এই কাজটি হত কার্ড পানশিং যন্ত্রের সাহায্যে।

বার সঙ্কেতের ব্যবহার হয় বই এবং অন্যান্য সামগ্রীর প্যাকেটের গায়ে। কতকগুলি সরু, মোটা, মাঝারি চওড়া ইত্যাদি দাগের সাহায্যে বার সঙ্কেত তৈরী হয়। বার সঙ্কেতের অর্থ পরিগণক যন্ত্র পড়তে পারে একপ্রকার অপটিক্যাল স্ক্যানার-এর মাধ্যমে।

OCR পদ্ধতিতে পরিগণক যন্ত্র ছাপা লেখা পড়তে পারে এবং সেই লেখাটিকে যথাযথ ভাবে স্মৃতি আধারে সঞ্চয় করে রাখতে পারে।

OMR পদ্ধতির সাথে প্রায় সবাই পরিচিত। বিভিন্ন প্রতিযোগিতামূলক পরীক্ষা এবং চাকুরীর পরীক্ষায় সঠিক উত্তর বাছাই পর্বে (Multiple Choice উত্তর পত্রে) কতকগুলি নির্দিষ্ট সংখ্যক ছোট ছোট চৌকো ঘর দেওয়া থাকে। প্রশ্নের সঠিক উত্তর পরিগণক যন্ত্রকে আগে থেকে জানানো থাকে। OMR এর কাজ হল পরীক্ষার্থীর উত্তরের সাথে পরিগণকের উত্তর কতগুলি মিলছে তার হিসাব করা।

ENTRANCE EXAMINATION

Roll No. **Centre**

☐☐☐☐☐☐ ☐☐☐☐

ANSWER SHEET

.....

Ques. No.	A	B	C	D	E
1.	☐	☐	☐	☐	☐
2.	☐	☐	☐	☐	☐
3.	☐	☐	☐	☐	☐
4.	☐	☐	☐	☐	☐
5.	☐	☐	☐	☐	☐

চিত্র ঃ OMR উত্তরপত্র

MICR পদ্ধতিটি ব্যবহার করা হয় ব্যাঙ্কের চেক্ বা ডিমাণ্ড ড্রাফট্ লিখতে ও পড়তে। লেখাটি হয় ম্যাগনেটিক কালির সাহায্যে। সাধারণ মানুষের ব্যবহৃত বর্ণমালা এবং সংখ্যা এই লেখাতে ব্যবহার করা হয়। এই পদ্ধতির সুবিধা হল—অল্প সময়ে অনেক তথ্য MIC Reader পড়ে নিতে পারে। এর ফলে টাইপ করে তথ্য সরবরাহ করতে হয়না এবং টাইপ সংক্রান্ত ভ্রান্তির কোন সুযোগ থাকে না।

তথ্য জ্ঞাপক একক ঃ

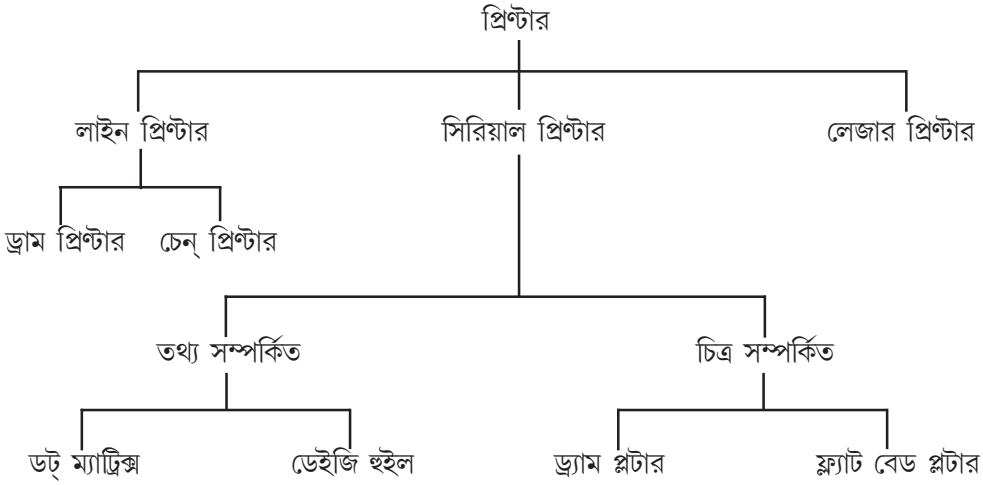
পরিগণক যন্ত্রে কোন প্রোগ্রাম চালানোর পর যে ফলাফল পাওয়া যায় তাকে ইংরাজীতে বলা হয় আউপুট। প্রকৃতিগতভাবে দু'ধরণের আউটপুট আমরা দেখতে পাই—(i) হার্ড কপি আউটপুট (ii) সফট্ কপি আউটপুট।

যে সমস্ত আউটপুট আমরা কাগজে ছাপা অবস্থায় পাই তাদের বলা হয় হার্ড কপি আউটপুট। অপরদিকে, যে সমস্ত আউটপুট আমরা পর্দায় দৃশ্যমান হতে দেখি অথবা শব্দ যন্ত্রের মাধ্যমে শুনতে পারি তাদের বলা হয় সফট্ কপি আউটপুট।

যে সমস্ত তথ্য জ্ঞাপনকারী একক সফট্ কপি আউটপুট তৈরী করে তাদের মধ্যে সবচেয়ে উল্লেখযোগ্য হল VDU বা Video Display Unit। এটি দেখতে টেলিভিশনের পর্দার মত এবং একে ইংরাজীতে বলা হয় মনিটর।

যে সমস্ত তথ্য জ্ঞাপনকারী একক হার্ড কপি আউটপুট তৈরী করে তাদের মধ্যে সবচেয়ে উল্লেখযোগ্য হল প্রিণ্টার বা ছাপাযন্ত্র।

নীচে বিভিন্ন প্রকার প্রিণ্টারের উল্লেখ করা হল।



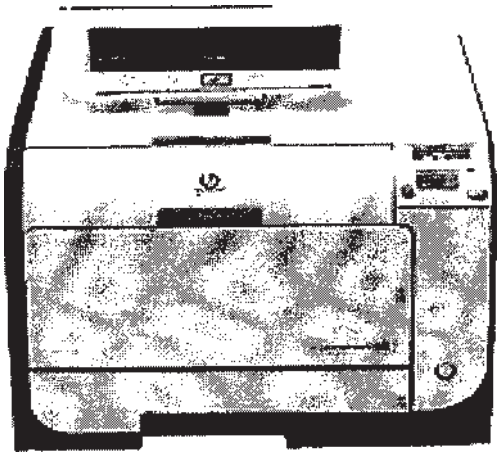
লাইন প্রিণ্টার একসাথে একটি করে লাইন ছাপতে পারে এবং ছাপার হরফ নানারকম হতে পারে।

ড্রাম প্রিণ্টারে একটি ড্রামের উপর বিভিন্ন হরফ খোদাই করা থাকে। ড্রামটি খুব স্পীডে ঘুরতে পারে। কয়েকটি হাতুড়ী ঐ ড্রামের উপর আঘাত করতে পারে। হাতুড়ী ও ড্রামের মাঝে থাকে একটি ফিতা এবং কাগজ। যে তথ্য ছাপা হবে তার সঙ্কেত অনুযায়ী হাতুড়ী যথাযথভাবে ড্রামের উপর আঘাত করে। হাতুড়ী এবং ড্রামের গতির মধ্যে অবশ্যই একটা সামঞ্জস্যবিধান করা থাকে।

সিরিয়াল প্রিণ্টার একবারে একটি করে অক্ষর ছাপতে পারে। সুতরাং, এই ধরণের প্রিণ্টার ধীর গতিতে ছাপতে পারে—প্রতি সেকেণ্ড 30-300 টি হরফ।

ডট্ ম্যাট্রিক্স হল একটি বহুল প্রচলিত প্রিণ্টার। এর প্রিণ্ট হেডে 9×7 ম্যাট্রিক্সের আকারে পিন্ সাজানো থাকে। কোন একটি হরফ ছাপতে হলে ঐ হরফের জন্য নির্দিষ্ট সঙ্কেত অনুযায়ী কতকগুলি পিন্ সক্রিয় হয়ে এগিয়ে আসে। এরপর প্রিণ্ট হেড ফিতের উপর চাপ দিলে ফিতের নীচে থাকা কাগজে হরফটি ছাপা হয়ে যায়। এই ধরনের প্রিণ্টার ইংরাজী ছাড়া অন্য ভাষার হরফও ছাপতে পারে। ডট্ ম্যাট্রিক্স প্রিণ্টারের দাম যথেষ্ট কম।

লেসার প্রিণ্টারে একটি লেসার বীম-এর সাহায্যে কাগজের উপর ছাপা হয়। যেটি ছাপা হবে তার একটি ছবি বিদ্যুৎশক্তি দ্বারা সঞ্চারিত করা হয় এবং অতিসূক্ষ্ম গুড়ো কালির দ্বারা কাগজে ছবি আঁকা হয়। এরপর কাগজটি একটি গরম রোলার-এর উপর দিয়ে ঘুরলে ছবি (বা লেখা) স্থায়ীভাবে ছাপা হয়ে যায়। এই ধরণের লেসার প্রিণ্টার বেশ দামী এবং এরা মিনিটে 20,000 টি লাইন পর্যন্ত ছাপতে পারে।



চিত্র-৩ : লেজার প্রিন্টার

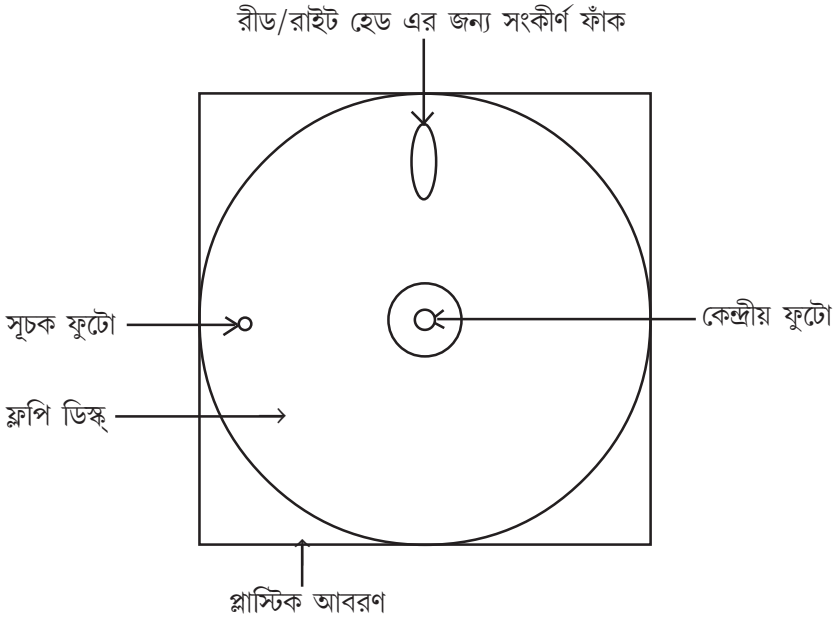
আনুষঙ্গিক স্মৃতি আধার ঃ

পরিগণকের কেন্দ্রীয় স্মৃতি আধারের শব্দ সঞ্চয়সীমা যেহেতু সীমিত সেহেতু বিপুল পরিমাণ তথ্য সঞ্চয় করে রাখতে আনুষঙ্গিক স্মৃতি আধার ব্যবহার করা হয়। যেমন—ফিক্সড্ ডিস্ক্, ফ্লপি ডিস্ক্, CD, পেন ড্রাইভ ইত্যাদি হল বিভিন্ন আনুষঙ্গিক স্মৃতি আধার। ফিক্সড্ ডিস্ক্ ব্যতীত অন্যান্য আনুষঙ্গিক স্মৃতি আধারগুলির সাহায্যে একটি পরিগণক যন্ত্র থেকে অন্য পরিগণক যন্ত্রে তথ্য এবং প্রোগ্রাম নিয়ে যাওয়া যেতে পারে।

ফিক্সড্ ডিস্ক্ হল পরিগণক যন্ত্রের সাথে স্থায়ীভাবে যুক্ত স্মৃতি আধার। PC-XT বা PC-AT নামের এই ধরণের ফিক্সড্ ডিস্কের তথ্যসঞ্চয় সীমা যথেষ্ট এবং স্পীডও যথেষ্ট বেশী।

ফ্লপি ডিস্ক্ হল পাতলা প্লাস্টিক জাতীয় উপাদানে তৈরী গোলাকার নমনীয় ডিস্ক্ যেটির উপর একটি চুম্বকীয় আস্তরণ থাকে। গোলাকার ডিস্ক্টি একটি শক্ত চৌকো খাপের মধ্যে রাখা থাকে। ডিস্ক্টিকে একটি ট্রে-তে রেখে ঘোরানো হয়। ট্রে-টিকে বলা হয় ফ্লপি ডিস্ক্ ড্রাইভ বা সংক্ষেপে FDD। ফ্লপি ডিস্ক্ বিভিন্ন মাপে পাওয়া যায়—

ইঞ্চি ইঞ্চি, 8 ইঞ্চি ব্যাসবিশিষ্ট। সর্বাধিক ব্যবহৃত হয় ইঞ্চি ব্যাস বিশিষ্ট ফ্লপি ডিস্ক্ যার তথ্যসঞ্চয় সীমা 1.44 MB। ফ্লপি ডিস্ক্ প্রথম বাজারে আনে IBM কোম্পানী 1970 সালে। বর্তমানে ফ্লপি ডিস্কের ব্যবহার অনেক কমে গেছে।



চিত্র : ফ্লপি ডিস্কের আভ্যন্তরীণ রূপ

আনুষঙ্গিক স্মৃতি আধার হিসাবে এখন সবচেয়ে বেশী ব্যবহৃত হয় Compact Disc বা সংক্ষেপে CD। এটি তৈরী হয় চুম্বকীয় আস্তরণযুক্ত শক্ত ধাতব গোলাকার পাত দ্বারা। এর ব্যাস 12 সেমি এবং তথ্যসঞ্চয় সীমা 700 MB বা 8.5 GB পর্যন্ত হতে পারে। কিছু CD তে শুধু একবারই তথ্য সঞ্চয় করা যায় এবং সঞ্চিত তথ্য যতবার খুশি পড়া যায়। এদের বলে Read-Only-CD। এছাড়া যে সমস্ত CD তে যতবার খুশি তথ্য সঞ্চয় করা যায় এবং অপ্রয়োজনে মুছে ফেলা যায় তাদের বলে Read-Write-CD। স্বভাবতঃই দ্বিতীয় প্রকার CD-র দাম তুলনামূলকভাবে বেশী।

বর্তমানে উচ্চ সঞ্চয়সীমা বিশিষ্ট পেন ড্রাইভের ব্যাপক প্রচলন শুরু হয়ে গেছে। একটি পেন ড্রাইভের তথ্যসঞ্চয়সীমা 1 GB থেকে 16 GB পর্যন্ত হতে পারে। তথ্যসঞ্চয়সীমা যত বেশী হবে পেন ড্রাইভের দামও তত বেশী হবে।

চিত্র-৪ : পেন ড্রাইভ

সংক্ষিপ্তসার ঃ

এই পাঠে পরিগণক যন্ত্র কি এবং ইহা আমাদের কি কি কাজ করতে পারে বা এই যন্ত্র দিয়ে আমরা কি কি কাজ করতে পারি তার আলোচনা করা হয়েছে। পৌরাণিক যুগ থেকে আধুনিক পরিগণক যন্ত্রের ঐতিহাসিক ক্রমবিকাশ বিশদভাবে আলোচিত হয়েছে। আধুনিক পরিগণক যন্ত্রের প্রকার ভেদ, যেমন মাইক্রো পরিগণক, মিনি পরিগণক, মেইনফ্রেম পরিগণক ও সুপার পরিগণক, সম্বন্ধে আলোচনা করা হয়েছে। আধুনিক পরিগণক যন্ত্রের বিভিন্ন অংশ, যেমন—কেন্দ্রীয় স্মৃতি আধার, যুক্তি ও গণনা কেন্দ্র, নিয়ন্ত্রণ কেন্দ্র, তথ্য গ্রহণ একক, তথ্য জ্ঞাপক একক এবং আনুষঙ্গিক স্মৃতি আধার সম্বন্ধে বিশদ বর্ণনা করা হয়েছে।

একক 2 ❑ তথ্য পরিবেশন Represention of Data

এককের উদ্দেশ্য ঃ

এই একক পাঠ করলে পাঠক নিম্নলিখিত বিষয়গুলি সম্বন্ধে জ্ঞাত হবে।

- **পরিগনক যন্ত্রে তথ্যসঞ্চয় ও তথ্য প্রক্রিয়া করণ।**
- **সংখ্যা প্রণালী দশমিক, দ্বিপ্রতীকী, অষ্টপ্রতীকী ও ষোড়শ প্রতীকী।**
- **বিভিন্ন সংখ্যা প্রনালীর ব্যবহার**
- **কোন একটি সংখ্যা প্রনালী থেকে অন্য সংখ্যা প্রনালীতে রূপান্তর।**
- **দ্বিপ্রতীকী সংখ্যার পাটিগনিত।**
- **নিধানপূরক সংখ্যা তন্ত্র**
- **দ্বিপ্রতীকী সংখ্যা প্রনালীতে পূরক সংখ্যার সাহায্যে বিয়োগ।**
- **দ্বিপ্রতীকী সংকেতলিপির সাহায্যে দশমিক সংখ্যার প্রকাশ।**

তথ্য পরিবেশন

পরিবেশনকে কিভাবে তথ্য সঞ্চয় এবং তথ্যের সংসাধন বা প্রক্রিয়াকরণ হয় তা বুঝতে হলে আমাদেরকে প্রথমে বিভিন্ন সংখ্যা প্রণালীর ভিত্তি ও গঠনকে জানতে হবে। আমাদের ব্যবহৃত সংখ্যা প্রণালীগুলির অধিকাংশই অবস্থানভিত্তিক। যেমন—দ্বিপ্রতীকী সংখ্যাপ্রণালী, অষ্টপ্রতীকী সংখ্যাপ্রণালী ইত্যাদি। অবস্থানভিত্তিক সংখ্যাপ্রণালীর ব্যতিক্রম হিসেবে রোমান সংখ্যাপ্রণালীর কথা বলা যেতে পারে। যেটিতে শূন্যমানের জন্য কোন প্রতীকের ব্যবহার নেই।

অবস্থানভিত্তিক সংখ্যাপ্রণালী ঃ যে কোন অবস্থানভিত্তিক সংখ্যাপ্রণালীতে নির্দিষ্ট সংখ্যক বিভিন্ন প্রতীক বা চিহ্ন ব্যবহার করা হয়। এই নির্দিষ্ট সংখ্যাটিকে বলা হয় সংখ্যাপ্রণালীর নিধান এবং বিভিন্ন প্রতীকগুলিই হল ঐ সংখ্যাপ্রণালীর মূল ভিত্তি বা অঙ্ক। যাদের সাহায্যে সংখ্যাপ্রণালীর যে কোন সংখ্যাকে প্রকাশ করা হয়। অতএব একটি সংখ্যাকে বলা যেতে পারে কয়েকটি চিহ্ন বা প্রতীকের সমাহার। সংখ্যায় ব্যবহৃত প্রতিটি প্রতীকের অবস্থানগত তাৎপর্য আছে যাকে স্থানাঙ্কমূল্য বলা হয়। সংখ্যায় ব্যবহৃত প্রতিটি প্রতীকের স্বীয় মূল্য এবং তার স্থানাঙ্কমূল্যের সাহায্যে সংখ্যাটির মান নির্ধারিত হয়। এটিই হল অবস্থানভিত্তিক সংখ্যাপ্রণালীর মূল কথা।

এবার আমরা কয়েকটি অবস্থানভিত্তিক সংখ্যাপ্রণালীর উদাহরণ পড়ব।

দশমিক সংখ্যাপ্রণালী ঃ সাধারণের পরিচিত এবং ব্যবহৃত সংখ্যা প্রণালী হল দশমিক সংখ্যাপ্রণালী। এই সংখ্যাপ্রণালীর নিধান 10 এবং মূল অঙ্ক (বা প্রতীক) গুলি হল 0, 1, 2, 3, 4, 5, 6, 7, 8, ও 9 ধনাত্মক বা ঋণাত্মক যে কোনো পূর্ণসংখ্যাকে আমরা প্রকাশ করি নিম্নলিখিত উপায়ে।

$\pm 2\ 4\ 5\ 1 \ = \ \pm\ (2\times10^3\ +\ 4\ \times10^2\ +\ 5\ \times10^1\ +\ 1\times\ 10^0)$

সহস্র	শতক	দশক	একক
অঙ্ক	অঙ্ক	অঙ্ক	অঙ্ক

এই সংখ্যাটিতে সর্বোচ্চ স্থানাঙ্ক মূল্য হল 2 প্রতীকটির যার স্থানাঙ্কমূল্য 2×10^3 এবং সর্বনিম্ন স্থানাঙ্কমূল্য হল 1 প্রতীকটির যার স্থানাঙ্কমূল্য 1×10^0

যেহেতু অবস্থানভিত্তিক সংখ্যাপ্রণালীতে যে কোন সংখ্যায় ব্যবহৃত প্রতিটি অঙ্কের নিজস্ব স্থানাঙ্কমূল্য নির্দিষ্ট, 123 সংখ্যাটি এবং 231 সংখ্যাটি কখনও অভিন্ন নয়, যদিও উভয় সংখ্যাতেই একই রকম অঙ্কের ব্যবহার করা হয়েছে।

দশমিক সংখ্যাপ্রণালীতে যে কোন বাস্তব সংখ্যাকে, উদাহরণস্বরূপ ধরা যাক ±5 274·608 কে আমরা প্রকার করি নিম্নলিখিত উপায়ে।

$\pm5274{\cdot}608\ =\ \pm(5\times10^3+2\times10^2+7\times10^1\ +\ 4\times10^0\ +\ 6\times10^{-1}+\ 0\times10^{-2}\ +\ 8\times10^{-3})$

এক দশাংশ এক শতাংশ এক সহস্রাংশ

এই উদাহরণটিতে সর্বনিম্ন স্থানাঙ্ক মূল্য রয়েছে ডান দিকের সর্বশেষ অঙ্কটির অর্থাৎ ৪-এর যার স্থানাঙ্ক মূল্য 8×10^{-3}। এই সংখ্যাটিতে 5274 এবং 608 এর মধ্যে ব্যবহৃত বিন্দুটি (·) দশমিক বিন্দু বলে পরিচিত।

কোন বাস্তব সংখ্যার উপরোক্ত উপায়ে সম্প্রসারণের ক্ষেত্রে ব্যবহৃত 10 এর ঘাত সীমিত সংখ্যক নাও হতে পারে। যেমন, $\sqrt{3}$ কে প্রকার করতে হবে এইভাবে—

$$\sqrt{3}\ =\ 1{\cdot}7320508075688...........$$

$$=\ 1\times10^0\ +\ 7\times10^{-1}\ +\ 3\ \times10^{-2}\ +\ 2\times10^{-3}+0\times10^{-4}+5\times10^{-5}+0\times10^{-6}$$

$$+8\times10^{-7}+0\times10^{-8}+7\times10^{-9}+5\times10^{-10}+6\times10^{-11}$$

$$+8\times10^{-12}+8\times10^{-13}+............$$

দ্বিপ্রতীকী সংখ্যা প্রণালী ঃ এই সংখ্যাপ্রণালীর নিধান 2 অতএব, মাত্র দুটি ভিন্ন প্রতীকের সাহায্যে যে কোন সংখ্যাকে প্রকাশ করতে হয় দ্বিপ্রতীকী সংখ্যা প্রণালীতে। এই প্রতীকদুটি হল 0 ও 1। এদেরকে বলা হয় দ্বিচলাঙ্ক বা ইংরেজীতে BIT (BInary digiT)।

দ্বিপ্রতীকী সংখ্যাপ্রণালীতে যে কোন সংখ্যাই হল কয়েকটি (অথবা অসীম সংখ্যক) 0 ও 1 এর সমাহার। উদাহরণ স্বরূপ, 1 0 1 1 0 1 এই দ্বিপ্রতীকী সংখ্যাটি এবং দশমিক সংখ্যাপ্রণালীর 45 সংখ্যাটি প্রকৃপক্ষে একই সংখ্যা—

101101 (দ্বিপ্রতীকী সংখ্যাপ্রণালীতে)

$=\ 1\times2^5\ +\ 0\times2^4\ +\ 1\times2^3\ +\ 1\times2^2\ +\ 0\times2^1\ +\ 1\times2^0$

$=32\ +\ 0\ +\ 8\ +\ 4\ +\ 0\ +\ 1$

= 45 (দশমিক সংখ্যা প্রণালীতে)

সংখ্যাপ্রণালীর নিধানকে ব্যবহার করে আমরা লিখতে পারি

$(101101)_2 = (45)_{10}$

দশমিক বিন্দুর মত, দ্বিপ্রতীকী সংখ্যাপ্রণালীতে ব্যবহার করা হয় অর্ধাংশবিন্দু। যেমন—

$$(101\cdot1101)_2 = 1\times2^2 + 0\times2^1 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-2} + 0\times2^{-3} + 1\times2^{-4}$$
$$= 4 + 0 + 1 + \cdot5 + \cdot25 + 0 + \cdot0625$$
$$= (5\cdot8125)_{10}$$

অষ্টপ্রতীকী সংখ্যাপ্রণালী ঃ এই সংখ্যাপ্রণালীর নিধান আট। অর্থাৎ, এই সংখ্যাপ্রণালীতে আটটি ভিন্ন প্রতীক ব্যবহার করা হয়। ব্যবহৃত আটটি প্রতীক হল 0, 1, 2, 3, 4, 5, 6, ও 7

উদাহরণ—

$$(5124)_8 = 5\times8^3 + 1\times8^2 + 2\times8^1 + 4\times8^0$$
$$= 2560 + 64 + 16 + 4$$
$$= (2644)_{10}$$
$$(137\cdot25)_8 = 1\times8^2+3\times8^1+7\times8^0+2\times8^{-1}+5\times8^{-2}$$
$$= 64 + 24 + 7 + 0\cdot25 + 0\cdot078125$$
$$= (95\cdot328125)_{10}$$

$(1\ 3\ 7\ \cdot25)_8$ এই অষ্ট প্রতীকী সংখ্যাতে ব্যবহৃত (·) বিন্দুটি হল অষ্টাংশবিন্দু।

ষোড়শ প্রতীকী সংখ্যাপ্রণালী ঃ এই সংখ্যাপ্রণালীর নিধান হল ষোল। যে ষোলটি ভিন্ন প্রতীক ব্যবহার করা হয় সেগুলি হল 0, 1, 2, 3, 4, 5, 6, 7 8, 9, A, B, C, D, E এবং F. দশমিক সংখ্যাপ্রণালীতে এদের মান যথাক্রমে 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 এবং 15।

উদাহরণ—

$$(5\ A\ 3\ 4\ F)_{16} = 5\times16^4 + 10\times16^3 + 3\times10^2 + 4\times10^1 + 15\times10^0$$
$$= 327680 + 40960 + 300 + 40 + 15$$
$$= (368995)_{10}$$

বিভিন্ন সংখ্যাপ্রণালীর ব্যবহার ঃ দশমিক সংখ্যাপ্রণালীর সাথেই আমরা সবচেয়ে বেশি পরিচিত এবং অভ্যস্ত রোমান সংখ্যা প্রণালী ও আমরা কখনও কখনও ব্যবহার করি। কিন্তু হিসেব বা গণনার প্রয়োজন হলেই আমরা দশমিক সংখ্যা প্রণালীতেই ফিরে যাই। এর কারণ হল দশমিক সংখ্যাপ্রণালীর নিধান হল দশ, যা খুব বেশী নয়, আবার কমও নয়। সংখ্যাপ্রণালীর নিধান যত কমতে থাকবে ততই সংখ্যা প্রকাশে অঙ্কসংখ্যা বাড়তে থাকবে। যেমন—দশমিক সংখ্যাপ্রণালীর 215 (অঙ্ক সংখ্যা তিন) সংখ্যাটি দ্বিপ্রতীকী সংখ্যায় প্রকাশ করলে দাঁড়াবে 11010111

(অঙ্কসংখ্যা আট) এবং ষোড়শ প্রতীকী সংখ্যাপ্রণালীতে এটি হবে D7 (অঙ্কসংখ্যা দুই)। সংখ্যাপ্রণালীর নিধান বেশী হলে সংখ্যার বিস্তার কমে ঠিকই। কিন্তু গণনার ক্ষেত্রে কিছুটা অসুবিধাজনক। এই সমস্ত কারণে দ্বিপ্রতীকী, অষ্টপ্রতীকী বা ষোড়ষ প্রতীকী সংখ্যা প্রণালী মানব মস্তিষ্ক দ্বারা গণনার পক্ষে খুব একটা সুবিধাজনক নয়। এরা বরং বৈদ্যুতিন্ পরিগণকের কর্মসহায়ক। দ্বিপ্রতীকী সংখ্যাপ্রণালী বৈদ্যুতিক মাধ্যমের পক্ষে বিশেষ স্মৃতিসহায়ক, কারণ দ্বিপ্রতীক সংখ্যাপ্রণালীতে যে দুটি ভিন্ন প্রতীক বা দ্বিচলাঙ্ক (0 এবং 1) ব্যবহার করা হয় তাদেরকে পরস্পরের বিপরীত বা পরিপূরক বলে ভাবা যায়। যেমন—'1' কে যদি ভাবা যায় 'উপস্থিত', তাহলে '0' এর অর্থ অবশ্যই 'অনুপস্থিত'; '1' কে যদি ধরি 'হ্যাঁ', তবে '0' এর মানে স্বাভাবিক ভাবেই 'না'। আমাদের ব্যবহৃত বিভিন্ন বৈদ্যুতিক যন্ত্র বা যন্ত্রাংশ দুটি ভিন্ন বৈদ্যুতিক অবস্থানে স্থিতিশীলভাবে থেকে কর্মসম্পাদন করে। যেমন—একটি বৈদ্যুতিক আলো হয় সপ্রভ নয়তো নিষ্প্রভ। এই দুটি ভিন্ন পরস্পর বিপরীত বৈদ্যুতিক অবস্থানকে বোঝাতে দুটি ভিন্ন প্রতীক '0' এবং '1' ব্যবহার করা যেতে পারে। বিপরীত দৃষ্টিভঙ্গি থেকে বলা যায় যে, '0' এবং '1' এই দুই দ্বিচলাঙ্ককে অনায়াসে সূচিত করা যেতে পারে দুটি ভিন্ন বৈদ্যুতিক অবস্থান (বৈদ্যুতিক voltage) এর সাহায্যে। এইজন্য বর্তমান বৈদ্যুতিন্ পরিগণকের গণনাকার্যে ব্যবহৃত হয় দ্বিপ্রতীকী সংখ্যাপ্রণালী। এই সংখ্যাপ্রণালীতে কিভাবে যোগ, বিয়োগ, গুণ এবং ভাগ করা হয় তার আলোচনা করা হয়েছে পরবর্তী অনুচ্ছেদে।

দ্বিপ্রতীকী সংখ্যা থেকে দশমিক সংখ্যায় রূপান্তর ঃ আমরা ইতিপূর্বেই দেখেছি, দ্বিপ্রতীকী সংখ্যায় ব্যবহৃত দ্বিচলাঙ্কগুলির স্থানাঙ্ক মূল্য ব্যবহার করে কিভাবে সংখ্যাটির সমমান দশমিক সংখ্যা পাওয়া যায়। নীচে আরও কিছু উদাহরণ দেওয়া হল।

উদাহরণ-1 :

$$(111001)_2 = 1\times 2^5 + 1\times 2^4 + 1\times 2^3 + 0\times 2^2 + 0\times 2^1 + 1\times 2^0$$
$$= (32 + 16 + 8 + 0 + 0 + 1)_0$$
$$= {}^{(57)}{}_{10}$$

উদাহরণ-2 :

$$(1011\cdot 101)_2 = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3}$$
$$= (8 + 0 + 2 + 1 + \cdot 5 + 0 + \cdot 125)_{10}$$
$$= (11\cdot 625)_{10}$$

উদাহরণ-3 :

$$(0\cdot 001101)_2 = 0\times 2^0 + 0\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3} + 1\times 2^{-4} + 0\times 2^{-5} + 1\times 2^{-6}$$
$$= (0 + 0 + 0 + 0\cdot 125 + 0\cdot 0625 + 0 + 0\cdot 015625)_{10}$$
$$= (0\cdot 203125)_{10}$$

দশমিক সংখ্যা থেকে দ্বিপ্রতীকী সংখ্যায় রূপান্তর ঃ এই রূপান্তর করা যায় একাধিক উপায়ে। নীচে দুটি পদ্ধতি আলোচনা করা হল।

প্রথম পদ্ধতি ঃ দশমিক সংখ্যাকে 2 এর বিভিন্ন ঘাতের সমষ্টিরূপে প্রকাশ করে কোন দশমিক সংখ্যার সমমান দ্বিপ্রতীকী সংখ্যা পাওয়া যায়। এই পদ্ধতিতে নীচের ছকটি বেশ উপযোগী।

.....	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	.....
.....	128	64	32	16	8	4	2	1	·5	·25	·125	·0625	·03125	.....

উদাহরণ-1 :

$(36)_{10} = (32 + 4)_{10}$

$= 2^5 + 2^2$

$= 1\times2^5 + 0 \times2^4 + 0\times2^3+1\times2^2 + 0\times2^1 + 0\times2^0$

$= (100100)_2$

উদাহরণ-2 :

$(171)_{10} = 128+32+8+2+1$

$= 2^6 + 2^5 + 2^3 + 2^1 + 2^0$

$= 1\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2+1\times2^1+1\times2^0$

$= (1101011)_2$

উদাহরণ-3 :

$(53\cdot625)_{10} = 32 + 16 + 4 + 1 + \cdot5 + \cdot125$

$= 2^5 + 2^4 + 2^2 + 2^0 + 2^{-1} + 2^{-3}$

$= 1\times2^5+1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$

$= (110101\cdot101)_2$

উদাহরণ-4 :

$(21\cdot8125)_{10} = 16+4+1+1+\cdot5 + \cdot25+\cdot0625$

$= 2^4 + 2^2 +2^0 + 2^{-1} + 2^{-2} +2^{-4}$

$= 1\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}+0\times2^{-3}+1\times2^{-4}$

$= (10101\cdot1101)_2$

দ্বিতীয় পদ্ধতি ঃ গুণ-ভাগ পদ্ধতি

এই পদ্ধতিতে কোন দশমিক সংখ্যাপ্রণালীর পূর্ণসংখ্যাকে তার সসমান দ্বিপ্রতীকী সংখ্যায় রূপান্তরিত করা হয় ভাগপ্রক্রিয়ার সাহায্যে। দশমিক সংখ্যাপ্রণালীতে কোন সংখ্যাকে 2 দ্বারা ভাগ করলে ভাগশেষ হয় 0 বা 1। ভাগফল যদি 0 এর বেশী হয় তাহলে সেই ভাগফলকে পুনরায় 2 দ্বারা ভাগ করতে হবে এবং ভাগশেষকে লিপিবদ্ধ করতে হবে। এই ভাগপ্রক্রিয়া ততক্ষণ চলতে থাকবে যতক্ষণ না ভাগফল 0 হচ্ছে। যেহেতু, দ্বিপ্রতীকী সংখ্যার সর্বাধিক সার্থক (most significant) দ্বিচলাঙ্কটি 2 এর সর্বোচ্চ ঘাত এর সাথে যুক্ত এবং সর্বাধিক অসার্থক দ্বিচলাঙ্কটি 2 এর সর্বনিম্ন ঘাত সাথে যুক্ত, ভাগপ্রক্রিয়ায় পাওয়া সর্বশেষ ভাগশেষটি হল দ্বিপ্রতীকী সংখ্যার সর্বাধিক সার্থক অঙ্ক এবং সর্বপ্রথম ভাগশেষটি দ্বিপ্রতীকী সংখ্যার সর্বনিম্ন সার্থক অঙ্ক।

উদাহরণ ঃ দশমিক সংখ্যাপ্রণালীর 183 সংখ্যাটিকে দ্বিপ্রতীকী সংখ্যায় রূপান্তর কর।

সমাধান ঃ

2	183	
2	91	1
2	45	1
2	22	1
2	11	0
2	5	1
2	4	1
2	2	0
2	1	0
	0	1

$\therefore (183)_{10} = (100110111)_2$

দ্বিপ্রতীকী সংখ্যাপ্রণালীতে কোন ভগ্নাংশ সংখ্যায় অর্ধাংশ বিন্দুর পরবর্তী দ্বিচলাঙ্কগুলি 2এর ঋণাত্মক ঘাত এর সাথে সম্পর্কযুক্ত। অতএবং ভগ্নাংশ-সংখ্যাটিকে 2 দ্বারা গুণ করলে গুণফলের পূর্ণাংশ হবে 0 অথবা 1। এই পূর্ণাংশটি হল অর্ধাংশবিন্দুর পরবর্তী প্রথম সার্থক অঙ্ক। গুণফলের পূর্ণাংশটিকে বাদ দিয়ে প্রাপ্ত ভগ্নাংশটিকে পুনরায় 2 দ্বারা গুণ করলে গুণফলের পূর্ণাংশটি হবে অর্ধাংশবিন্দুর পরবর্তী দ্বিতীয় সার্থক অঙ্ক। এই গুণপ্রক্রিয়া ক্রমাণ্বয়ে চালিয়ে গেলে অর্ধাংশবিন্দুর পরবর্তী দ্বিচলাঙ্ক গুলিকে পাওয়া যাবে। এইভাবে গুণপ্রক্রিয়ায় প্রাপ্ত দ্বিপ্রতীকী ভগ্নাংশটি সসীম না হয়ে আবৃত্ত ভগ্নাংশ বা পৌনঃপুনিক ক্রমিক ভগ্নাংশ হতে পারে।

উদাহরণ-1 : $(\cdot 145)_{10}$ কে দ্বিপ্রতীকী সংখ্যায় রূপান্তর কর।

সমাধান ঃ

দশমিক ভগ্নাংশ	**2 দ্বারা গুণ**	**গুণফলের পূর্ণাংশ**
0·145	·145 × 2 = ·290	0 (প্রথম সার্থক অঙ্ক)
0·29	·29 × 2 = ·58	0
0·58	·58 × 2 = 1·16	1
0·16	·16 × 2 = ·32	0
0·32	·32 × 2 = ·64	0
0·64	·64 × 2 = 1·28	1
0·28	·28 × 2 = ·56	0
0·56	·56 × 2 = 1·12	1
0·12	·12 × 2 = ·24	0
.	.	.
.	.	.
.	.	.
.	.	.

$\therefore (\cdot 145)_{10} = (\cdot 001001010.....)_2$

উদাহরণ-2 : $(0{\cdot}6875)_{10}$ কে দ্বিপ্রতীকী ভগ্নাংশে রূপান্তর কর।

সমাধান ঃ

দশমিক ভগ্নাংশ	2 দ্বারা গুণ	গুণফলের পূর্ণাংশ
0·6875	·6875 × 2 = 1·3750	1 (প্রথম সার্থক অঙ্ক)
0·375	·375 × 2 = ·750	0
0·75	·75 × 2 = 1·50	1
0·5	·5 × 2 = 1·0	1

$\therefore (0{\cdot}6875)_{10} = (0{\cdot}1011)_2$

উদাহরণ-3 : $(0{\cdot}9)_{10}$ কে দ্বিপ্রতীকী ভগ্নাংশে রূপান্তর কর।

সমাধান ঃ

দশমিক ভগ্নাংশ	2 দ্বারা গুণ	গুণফলের পূর্ণাংশ
0·9	·9 × 2 = 1·8	1 (প্রথম সার্থক অঙ্ক)
0·8	·8 × 2 = 1·6	1
0·6	·6 × 2 = 1·2	1
0·2	·2 × 2 = 0·4	0
0·4	·4 × 2 = 0·8	0
0·8	·8 × 2 = 1·6	1
0·6	·6 × 2 = 1·2	1
.	.	.
.	.	.
.	.	.
.	.	.

$\therefore (0{\cdot}9)_{10} = (\cdot 11100\ (1100))_2$

আবৃত্ত

উদাহরণ-4 : $(34{\cdot}875)_{10}$ কে দ্বিপ্রতীকী সংখ্যায় রূপান্তর করো।

সমাধান ঃ প্রথমে প্রদত্ত সংখ্যাটির পূর্ণাংশকে ভাগ প্রক্রিয়ার সাহায্যে সমমান দ্বিপ্রতীকী সংখ্যায় রূপান্তর করা যাক—

		ভাগশেষ
2	34	
2	17	0
2	8	1
2	4	0
2	2	0
2	1	0
	0	1 .....(সর্বাধিক সার্থক অঙ্ক)

$\therefore (34)_{10} = (\ 1\ 00010)_2$

এবার প্রদত্ত সংখ্যাটির ভগ্নাংশকে অর্থাৎ 0·875 কে গুণপ্রক্রিয়ার সাহায্যে সমমান দ্বিপ্রতীকী ভগ্নাংশে রূপান্তর করা যাক।

দশমিক ভগ্নাংশ	2 দ্বারা গুণ	গুণফলের পূর্ণাংশ
0·875	·875 × 2 = 1·750	1(অর্ধাংশ বিন্দুর পর প্রথম সার্থক অঙ্ক)
0·75	·75 × 2 = 1·50	1
0·5	·5 × 2 = 1·0	1

∴ $(0·875)_{10} = (0·111)_2$

∴ প্রদত্ত সংখ্যাটিকে দ্বিপ্রতীকী সংখ্যায় রূপান্তর করলে দাঁড়ায়

$(34·875)_{10} = (100010·111)_2$

দশমিক সংখ্যা প্রণালী থেকে অষ্টপ্রতীকী বা ষোড়ষ প্রতীকী সংখ্যাপ্রণালীতে রূপান্তর ঃ

দশমিক সংখ্যা প্রণালী থেকে যেভাবে দ্বিপ্রতীকী সংখ্যাপ্রণালীতে রূপান্তর করা হয়েছে উপরের অনুচ্ছেদে ঠিক একই নিয়মে দশমিক সংখ্যাপ্রণালী থেকে অষ্টপ্রতীকী বা ষোড়ষ প্রতীকী সংখ্যা প্রণালীতে রূপান্তর করা যায়। তবে এক্ষেত্রে ২ দ্বারা গুণ/ভাগ এর পরিবর্তে নিধান সংখ্যা 8 (অষ্ট প্রতীকী সংখ্যা প্রণালীর ক্ষেত্রে) বা নিধানসংখ্যা 16 (ষোড়শ প্রতীকী সংখ্যাপ্রণালীর ক্ষেত্রে) দ্বারা গুণভাগ করতে হবে। নীচে এই রূপান্তরের কয়েকটি উদাহরণ দেওয়া হল।

উদাহরণ-1 : $(215)_{10}$ সংখ্যাটিকে অষ্টপ্রতীকী সংখ্যায় রূপান্তর কর।

সমাধান ঃ

		ভাগশেষ
8	215	
8	26	7
8	3	2
	0	3

∴ $(215)_{10} = (327)_8$

উদাহরণ-2 : $(317·45)_{10}$ সংখ্যাটিকে অষ্টপ্রতীকী সংখ্যায় রূপান্তর কর।

সমাধান ঃ প্রথমে $(317)_{10}$ কে অষ্টপ্রতীকী সংখ্যায় রূপান্তর করা হল—

		ভাগশেষ
8	317	
8	39	5
8	4	7
	0	4

∴ $(317)_{10} = (475)_8$

এবার $(0{\cdot}45)_{10}$ কে অষ্ট প্রতীকী ভগ্নাংশে রূপান্তর করা হল—

দশমিক ভগ্নাংশ	8 দ্বারা গুণ	গুণফলের পূর্ণাংশ
0·45	·45 × 8 = 3·60	3
0·6	·6 × 8 = ·48	0
0·48	·48 × 8 = 3·84	3
0·84	·84 × 8 = 6·72	6
0·72	·72 × 8 = 5·76	5
0·76	·76 × 8 = 6·08	6
0·08	·08 × 8 = ·64	0
0·64	·64 × 8 = 5·12	5
.	.	.
.	.	.
.	.	.

$\therefore (0{\cdot}45)_{10} = (0{\cdot}30365605....)_8$

$\therefore$ প্রদত্ত সংখ্যাটির সমমান অষ্টপ্রতীকী সংখ্যাটি হল

$(317{\cdot}45)_{10} = (475{\cdot}30365605.....)_8$

দ্রষ্টব্য ঃ কোন সংখ্যাপ্রণালীতে কোনও একটি সংখ্যায় ভগ্নাংশবিন্দুর পর সসীম সংখ্যক সার্থক অঙ্ক থাকলেও অন্য কোন সংখ্যাপ্রণালীতে তা সসীম সংখ্যক নাও থাকতে পারে।

উদাহরণ-3 : $(2179{\cdot}125)_{10}$ সংখ্যাটিকে ষোড়শ প্রতীকী সংখ্যায় রূপান্তর কর।

সমাধান ঃ প্রথমে প্রদত্ত সংখ্যাটি পূর্ণাংশ অর্থাৎ $(2179)_{10}$ কে সমমান ষোড়শ প্রতীকী সংখ্যায় রূপান্তর করা হল ভাগপ্রক্রিয়ার সাহায্যে

		ভাগশেষ
16	2179	
16	136	3
16	8	8
	0	8

$\therefore (2178)_{10} = (883)_{16}$

এবার প্রদত্ত সংখ্যাটির ভগ্নাংশ অর্থাৎ $(\cdot 125)_{10}$ কে সমমান ষোড়ষ প্রতীকী ভগ্নাংশে রূপান্তর করা হল গুণপ্রক্রিয়ার সাহায্যে

দশমিক ভগ্নাংশ	16 দ্বারা গুণ	গুণফলের পূর্ণাংশ
·125	·125 × 16 = 2·000	2

$\therefore (\cdot 125)_{10} = (\cdot 2)_{16}$

$\therefore$ প্রদত্ত সংখ্যাটির সমমান ষোড়শ প্রতীকী সংখ্যাটি হল $(883{\cdot}2)_{16}$

উদাহরণ-4 : $(5274{\cdot}58)_{10}$ সংখ্যাটিকে ষোড়ষ প্রতীকী সংখ্যায় পরিণত কর।

সমাধান ঃ প্রথমে ভাগপ্রক্রিয়ার সাহায্যে $(5274)_{10}$ এর সমমান ষোড়শ প্রতীকী সংখ্যা বের করা যাক।

		ভাগশেষ
16	5274	
16	329	10 (=A)
16	20	9
16	1	4
	0	1

∴ $(5274)_{10} = (149A)_{16}$

এবার $(\cdot 58)_{10}$ কে গুণপ্রক্রিয়া দ্বারা সমমান ষোড়শ প্রতীকী ভগ্নাংশে পরিণত করা যাক।

16 দ্বারা গুণ	**গুণফলের পূর্ণাংশ**
·58 × 16 = 9·28	9
·28 × 16 = 4·48	4
·48 × 16 = 7·68	7
·68 × 16 = 10·88	10 (= A)
·88 × 16 = 14·08	14 (= E)
·08 × 16 = 1·28	1
·28 × 16 = 4·48	4
.	.
.	.
.	.

∴ $(\cdot 58)_{10} = (\cdot 9\ 4\ 7\ A\ E\ 1\ (4\ 7\ A\ E\ 1)\ )_{16}$

দ্বিপ্রতীকী সংখ্যাপ্রণালী থেকে অষ্টপ্রতীকী সংখ্যাপ্রণালীতে রূপান্তর ঃ

অষ্টপ্রতীকী সংখ্যাপ্রণালীর আটটি ভিন্ন প্রতীক এর সমমান দ্বিপ্রতীকী সংখ্যাগুলি নীচের টেবিলে দেওয়া হল।

অষ্টপ্রতীকী সংখ্যা	সমমান	দ্বিপ্রতীকী সংখ্যা	তিন অঙ্কে প্রকাশ
0	0		000
1	1		001
2	10		010
3	11		011
4	100		100
5	101		101
6	110		110
7	111		111

উপরের টেবিলটি ব্যবহার করে খুব সহজেই দ্বিপ্রতীকী সংখ্যা থেকে সমমান অষ্ট প্রতীকী সংখ্যা পাওয়া যেতে পারে। এজন্য প্রথমে দ্বিপ্রতীকী সংখ্যায় ব্যবহৃত দ্বিচলাঙ্কগুলির মধ্যে দল ভাগ করতে হবে। প্রতিটি দলে তিনটি করে দ্বিচলাঙ্ক থাকবে। দ্বিপ্রতীকী সংখ্যাটির পূর্ণাংশের জন্য এই দলভাগ শুরু হবে অর্ধবিন্দু থেকে বামদিকে অর্থাৎ

সর্বনিম্ন সার্থক অঙ্ক থেকে শুরু করে ক্রমশঃ সর্বাধিক সার্থক অঙ্কের দিকে এগোতে হবে এবং প্রয়োজনে সর্বাধিক সার্থক অঙ্কের বামদিকে 0-এর উপস্থিতি ধরে নিতে হবে। এতে সংখ্যাটির মানের কোন পরিবর্তন হয় না। অর্ধাংশবিন্দুর ডানদিকের দ্বিচলাঙ্কগুলির জন্য দলভাগ শুরু করতে হবে অর্ধাংশবিন্দুর ঠিক পরবর্তী প্রথম সার্থক অঙ্ক থেকে। এক্ষেত্রে, প্রয়োজনমত দ্বিপ্রতীকী সংখ্যার ভগ্নাংশটির পরে শূন্যের উপস্থিতি ধরা যেতে পারে যাতে প্রতিটি দলে তিনটি করে দ্বিচলাঙ্ক থাকে। এইভাবে দলবিভাজনের পর প্রতিটি দলের সমমান অষ্টপ্রতীকী সংখ্যা টেবিল থেকে দেখে পরপর বসিয়ে দিলেই সমমান অষ্টপ্রতীকী সংখ্যা পাওয়া যাবে।

উদাহরণ ঃ নীচের সংখ্যাগুলির সমমান অষ্টপ্রতীকী সংখ্যা উদ্ধার কর।

(ক) $(1101001)_2$

(খ) $(1011{\cdot}01)_2$

(গ) $(110{\cdot}1011011)_2$

সমাধান ঃ

(ক) $(1101001)_2$

$= \underbrace{001}\ \underbrace{101}\ \underbrace{001}$ (বামদিকে দুটি 0 বসিয়ে)

$= (1\ \ 5\ \ 1)_8$

$= (151)_8$

(খ) $(1011 \cdot 01)_2$

$= \underbrace{001}\ \underbrace{011} \cdot \underbrace{010}$ (বামদিকে দুটি 0 এবং ডানদিকে একটি 0 বসিয়ে)

$= (1\ \ 3 \cdot 2)_8$

$= (13{\cdot}2)_8$

(গ) $(110{\cdot}1011011)_2$

$= (\underbrace{110} \cdot \underbrace{101}\ \underbrace{101}\ \underbrace{100})_2$ (ডানদিকে দুটি 0 বসিয়ে)

$= (6{\cdot}554)_8$

অষ্ট প্রতীকী সংখ্যা থেকে দ্বিপ্রতীকী সংখ্যায় রূপান্তর ঃ

এই রূপান্তর কার্যেও উপরের টেবিলটি অত্যন্ত সহায়ক। অষ্টপ্রতীকী সংখ্যার প্রতিটি অঙ্কের সমমান দ্বিপ্রতীকী সংখ্যা পরপর লিখে গেলে অষ্টপ্রতীকী সংখ্যাটির সমমান দ্বিপ্রতীকী সংখ্যা পাওয়া যায়।

উদাহরণ ঃ নীচের অষ্টপ্রতীকী সংখ্যাগুলিকে দ্বিপ্রতীকী সংখ্যায় রূপান্তর কর ঃ

(ক) $(4\ 3\ 1)_8$

(খ) $(2\ 6\ 5 \cdot 7\ 3)_8$

সমাধান ঃ $(4\ 3\ 1)_8$

(ক) $= 4\quad 3\quad 1$

$= \overbrace{100}\quad \overbrace{011}\quad \overbrace{001}$

$= (100\quad 011\quad 001)_2$

(খ) $(265 \cdot 73)_8$

$= 2 \quad 6 \quad 5 \quad \cdot \quad 7 \quad 3$

$= \overbrace{010} \quad \overbrace{110} \quad \overbrace{101} \quad 111 \quad 011$

$= (10 \quad 110 \quad 101 \cdot 111011)_2$

দ্বিপ্রতীকী সংখ্যাপ্রণালী এবং ষোড়শ প্রতীকী সংখ্যা প্রণালীর মধ্যে পারস্পরিক রূপান্তর ঃ

ষোড়শপ্রতীকী সংখ্যা প্রণালীর ষোলটি ভিন্ন প্রতীককে চার অঙ্কের দ্বিপ্রতীকী সংখ্যাদ্বারা সুনির্দিষ্টভাবে প্রকাশ করা যায়, যেহেতু $2^4 = 16$ । এই সুনির্দিষ্ট নিয়মটি নীচের টেবিলে দেওয়া হল।

ষোড়শপ্রতীকী সংখ্যা	সমমান দশমিক সংখ্যা	সমমান দ্বিপ্রতীকী সংখ্যা
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

উপরের টেবলটি ব্যবহার করে খুব সহজেই দ্বিপ্রতীকী সংখ্যাপ্রণালী এবং ষোড়শপ্রতীকী সংখ্যাপ্রণালীর মধ্যে পারস্পরিক রূপান্তর সম্ভব। এই রূপান্তরের নিয়ম পূর্ববর্তী অনুচ্ছেদে ব্যবহৃত নিয়মের মত। তবে এক্ষেত্রে সংখ্যায় ব্যবহৃত দ্বিচলাঙ্কগুলিকে দলভাগ করার সময় প্রতিটি দলে চারটি করে দ্বিচলাঙ্ককে একসাথে রাখতে হবে। নীচে এই রূপান্তরের উদাহরণ দেওয়া হল।

উদাহরণ ঃ নীচের দ্বিপ্রতীকী সংখ্যাগুলিকে ষোড়শপ্রতীকী সংখ্যায় রূপান্তর কর।

(ক) $(10111011001)_2$

(খ) $(1110001011 \cdot 001001)_2$

সমাধান ঃ

(ক) $(101110 \; 1100)_2$

= 0101 1101 1001 (বামদিকে একটি 0 বসিয়ে)

| | |

= 5 D 9

= $(5D9)_{16}$

(খ) $(1110001011 \cdot 001001)_2$

= 0011 1000 1011 · 0010 0100 (বামদিকে ও ডানদিকে

| | | | | 0 দুটি করে বসিয়ে)

3 8 B · 2 4

= $(38B \cdot 24)_{16}$

উদাহরণ ঃ নীচের ষোড়শ প্রতীকী সংখ্যাদুটির সমমান দ্বিপ্রতীকী সংখ্যা বের কর—

(ক) $(F42)_{16}$ (খ) $(1C\ 3\ \cdot 5B)_{16}$

সমাধান ঃ

(ক) $(F\ 4\ 2)_{16}$

= $(\ 1111\ \ 0100\ \ 0010)_2$

= $(111101000010)_2$

(খ) = $(1C3\ \cdot 5B)_{16}$

= 1 C 3 · 5 B

0001 1100 0011 · 0101 1011

= $(111000011 \cdot 01011011)_2$

দ্বিপ্রতীকী পাটিগণিত ঃ আমরা সাধারণ ব্যবহারিক জীবনে হিসাব বা গণনাসংক্রান্ত কাজ দশমিক সংখ্যাপ্রণালীতে করতে অভ্যস্ত। কিন্তু পরিগণকের আভ্যন্তরীণ তথ্য বিন্যাস এবং তথ্যের সংসাধন হয় দ্বিপ্রতীকী সংখ্যা প্রণালী এবং দ্বিপ্রতীকী পাটিগণিতের সাহায্যে। কাজেই পরিগণকের আভ্যন্তরীণ কার্যকলাপ বুঝতে হলে বা নিয়ন্ত্রণ করতে হলে দ্বিপ্রতীকী সংখ্যাপ্রণালী এবং সেই সঙ্গে দ্বিপ্রতীকী পাটিগণিতও জানা প্রয়োজন। এই অনুচ্ছেদটিতে আমরা দ্বিপ্রতীকী সংখ্যাপ্রণালীতে যোগ, বিয়োগ, গুণ ও ভাগ করতে শিখব।

দ্বিপ্রতীকী যোগফল ঃ

দুটি দ্বিচলাঙ্ককে যোগ করার নিয়ম নীচের ছকে দেওয়া হল ঃ

অঙ্ক যোজ্য	যোজক অঙ্ক	গাণিতিক রাশিমালা	সমষ্টি অঙ্ক	বাহিত অঙ্ক
0	0	0 + 0	0	0
0	1	0 + 1	1	0
1	0	1 + 0	1	0
1	1	1 +1	0	1

এই ছকটিকে বলা হয় অর্ধযোজক টেবিল। দশমিক সংখ্যাপ্রণালীতে 9 এর সাথে যখন 1 যোগ করা হয় তখন যোগফলকে প্রকাশ করা 10 সংখ্যা দ্বারা যার দ্বিতীয় অঙ্কটি, অর্থাৎ 1 হল বাহিত অঙ্ক। একই নিয়মে দ্বিপ্রতীকী সংখ্যা প্রণালীতে যখন 1 এর সাথে 1 যোগ করা হয় তখন সমষ্টি সংখ্যাকে লেখা হয় 10 $(=(10)_2)$. এর দ্বিতীয় অঙ্কটি, অর্থাৎ 1 হল বাহিত অঙ্ক। দুটি দ্বিপ্রতীকী সংখ্যাকে যোগ করার সময় এই বাহিত অঙ্কটি পরবর্তী যোজ্য সংখ্যাদুটির সাথে যোগ করতে হবে। অর্থাৎ, এক্ষেত্রে তিনটি দ্বিচলাঙ্ককে একবারে যোগ করতে হবে। এই কাজের জন্য নীচের পূর্ণযোজক টেবিলটি বিশেষ উপযোগী।

যোজ্য অঙ্ক	যোজক অঙ্ক	বাহিত অঙ্ক	সমষ্টি অঙ্ক	পরবর্তী বাহিত অঙ্ক
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

অর্ধযোজক ও পূর্ণযোজক টেবিলদুটি ব্যবহার করে যে কোন দুটি দ্বিপ্রতীকী সংখ্যার যোগফল বার করা যায়।

উদাহরণ ঃ মান নির্ণয় করো

$(1\ 0\ 1\ 1\ 0 \cdot 1\ 1)_2 + (1\ 1\ 1 \cdot 0\ 1)_2$

সমাধান ঃ

বাহিত অঙ্ক	1 1
যোজ্য সংখ্যা	1 0 1 1 0 · 1 0
যোজক সংখ্যা	+ 1 1 1 · 0 1
দ্বিপ্রতীকী যোগফল	1 1 1 0 1 · 1 1

উদাহরণ ঃ দ্বিপ্রতীকী যোগফল বের কর —

$(1\ 0\ 1\ 1\ 0)_2 + (1\ 1\ 0\ 1\ 0)_2$

সমাধান ঃ

বাহিত অঙ্ক	1 1 1
যোজ্য সংখ্যা	1 0 1 1 0
যোজক সংখ্যা	1 1 0 1 0
দ্বিপ্রতীকী যোগফল	

দ্বিপ্রতীকী গুণফল ঃ 1 1 0 0 0 0

দুটি দ্বিচলাঙ্কের গুণফলের নিয়ম দশমিক সংখ্যাপ্রণালীর 0 এবং 1 এর গুণফলের মত। নীচের টেবিলটিতে তা দেখালো হল—

গুণনীয়	গুণক	গাণিতিক রাশিমালা	গুণফল
0	0	0 × 0	0
0	1	0 × 1	0
1	0	1 × 0	0
1	1	1 × 1	1

এই ছকটি ব্যবহার করে এবং দ্বিপ্রতীকী যোগপদ্ধতির সাহায্যে যে কোন দুটি দ্বিপ্রতীকী সংখ্যার গুণফল নির্ণয় করা যায়।

উদাহরণ ঃ দ্বিপ্রতীকী গুণফল নির্ণয় করো ঃ

(ক) 1 1 0 1 0 × 1 0 1

(খ) 1 0 1 1 · 0 1 1 × 1 1 0 · 1 1

সমাধান ঃ

1 1 0 1 0	
× 1 0 1	
1 1 0 1 0	(প্রথম আংশিক গুণফল)
0 0 0 0 0 ×	(দ্বিতীয় আংশিক গুণফল)
1 1 0 1 0 ×	(তৃতীয় আংশিক গুণফল)
1 0 0 1 1 1 0	(নির্ণেয় গুণফল)

(খ)

1 0 1 1 · 0 1 1	
× 1 1 0 · 1 1	
1 0 1 1 0 1 1	(প্রথম আংশিক গুণফল)
1 0 1 1 0 1 1 ×	(দ্বিতীয় আংশিক গুণফল)
1 0 0 0 1 0 0 0 1	(প্রথম আংশিক যোগফল)
0 0 0 0 0 0 0 × ×	(তৃতীয় আংশিক গুণফল)
1 0 0 0 1 0 0 0 1	(দ্বিতীয় আংশিক যোগফল)
1 0 1 1 0 1 1 × × ×	(চতুর্থ আংশিক গুণফল)
1 1 1 1 1 0 1 0 0 1	(তৃতীয় আংশিক যোগফল)
1 0 1 1 0 1 1 × × × ×	(পঞ্চম আংশিক গুণফল)
1 0 0 1 1 0 0 ·1 1 0 0 1	(নির্ণেয় গুণফল)

যেহেতু গুণনীয় সংখ্যাটিতে অর্ধাংশবিন্দুর পর তিনটি দ্বিচলাঙ্ক এবং গুণক সংখ্যাটিতে অর্ধাংশবিন্দুর পর দুইটি দ্বিচলাঙ্ক আছে

গুণফলটিতে অর্ধবিন্দুর পর (3+2=) 5টি দ্বিচলাঙ্ক থাকবে।

অর্থাৎ, অর্ধাংশবিন্দুর অবস্থান নির্ণয় পদ্ধতি দশমিক সংখ্যাপ্রণালীর মতই।

দ্বিপ্রতীকী বিয়োগফল ঃ

দুটি দ্বিচলাঙ্কের বিয়োগফল নির্ণয়ের জন্য নীচের টেবিলটি ব্যবহার করা যেতে পারে

বিযোজ্য অঙ্ক	বিয়োজক অঙ্ক	গাণিতিক রাশিমালা	বিয়োগফল	গৃহীত অঙ্ক
0	0	0 – 0	0	0
1	0	1 – 0	1	0
0	1	0 – 1	1	1
1	1	1 – 1	0	0

এই ছকটিকে বলা হয় অর্ধ বিযোজক টেবিল। পূর্ণ বিযোজক টেবিলটি নীচে দেওয়া হল

বিযোজ্য অঙ্ক	বিযোজক অঙ্ক	গৃহীত অঙ্ক	বিয়োগফল	পরবর্তী গৃহীত অঙ্ক
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

অর্ধবিযোজক এবং পূর্ণবিযোজক টেবিল ব্যবহার করে যে কোন দুটি দ্বিপ্রতীকী বিয়োগফল নির্ণয় করা যেতে পারে।

উদাহরণ ঃ বিয়োগফল নির্ণয় কর—

(ক) 1 1 0 1 · 0 1 1 – 1 1 · 1 0

(খ) 1 1 0 · 1 0 1 – 1 0 1 1 · 1 1 0 1

সমাধান ঃ

(ক) গৃহীত অঙ্ক 1 1

বিযোজ্য সংখ্যা 1 1 0 1 · 0 1 1

বিযোজক সংখ্যা – 1 1 · 1 0 1 ডানদিকে একটি 0 বসিয়ে

বিয়োগফল 1 1 0 1 · 1 1 1

(খ) যেহেতু 1 0 1 1 · 1 1 0 1 সংখ্যাটি 1 1 0 · 1 0 1 অপেক্ষা বৃহত্তর

প্রথমে 1 0 1 1 · 1 1 0 1 থেকে 1 1 0 · 1 0 1 কে বিয়োগ করা হল

$$
\begin{array}{r}
1\,0\,1\,1 \cdot 1\,1\,0\,1 \\
-\ 1\,1\,0 \cdot 1\,0\,1\,0 \\
\hline
1\,0\,1 \cdot 0\,0\,1\,1
\end{array}
$$

(ডানদিকে একটি 0 বসিয়ে)

∴ নির্ণেয় বিয়োগফল = – 101·0011

উপরে বর্ণিত দ্বিপ্রতীকী বিয়োগফল নির্ণয় পদ্ধতিটি পরিগণকের কার্যসম্পাদনের পক্ষে বিশেষ উপযোগী নয়। বরং সেক্ষেত্রে দুটি দ্বিপ্রতীকী সংখ্যার বিয়োগফলকে একটি দ্বিপ্রতীকী সংখ্যার সাথে অপর দ্বিপ্রতীকী সংখ্যাটির ঋণাত্মক মানের যোগফল হিসাবে ধরা যেতে পারে। এর ফলে একই তড়িৎবর্তনীর সাহায্যে দুটি দ্বিপ্রতীকী সংখ্যার যোগফল ও বিয়োগফল নির্ণয় সম্ভব। তবে এর জন্য প্রতিটি দ্বিপ্রতীকী সংখ্যার প্রকৃত চিহ্নকে ('+' অথবা '–') পরিগণকের অভ্যন্তরীণ তথ্যবিন্যাসে সংযোজিত করা দরকার। এই চিহ্নিত করণের একটি উপায় হল সংখ্যাটির বামদিকে (অর্থাৎ সর্বাধিক সার্থক অঙ্কের বামদিকে) একটি অতিরিক্ত দ্বিচলাঙ্ক সংযোজন করা। '+' চিহ্ন বোঝাতে 0 এবং '–' চিহ্ন বোঝাতে 1 ব্যবহার করা যেতে পারে। উদাহরণস্বরূপ 14 সংখ্যাটিকে লিখতে হবে 0, 1110 এবং –9 সংখ্যাটিকে লিখতে হবে, 1, 001, (এখানে ',' ব্যবহার করা হয়েছে শুধু মাত্র পাঠকের বোঝার সুবিধার্থে)। এই নিয়মানুযায়ী প্রথম দ্বিচলাঙ্কটি হল চিহ্ন নির্দেশক দ্বিচলাঙ্ক।

পূরক সংখ্যাতন্ত্র ঃ

ঋণাত্মক সংখ্যা প্রকাশের প্রচলিত রীতি হল পূরক সংখ্যাতন্ত্রের ব্যবহার। পূরক সংখ্যাতন্ত্র দু'ধরনের হতে পারে ঃ (ক) নিধান পূরক সংখ্যাতন্ত্র (Radix complement number system),

(খ) ঊণনিধান পূরক সংখ্যাতন্ত্র (Radix-minus one complement number system)

(ক) নিধানপূরক সংখ্যাতন্ত্র ঃ দশমিক সংখ্যাপ্রণালীর নিধান দশ। এই সংখ্যাপ্রণালীতে *m* সংখ্যক অঙ্ক বিশিষ্ট কোন সংখ্যার (N) নিধানপূরক সংখ্যাটি হল $(10^m–N)$. উদাহরণস্বরূপ, 1547 সংখ্যাটির নিধানপূরক সংখ্যা হল $(10^4–1547)$ = 10000 – 1547 = 8453।

যেহেতু $(10^m–N)$ কে লেখা যায় নিম্নরূপে

$$10^m - N = \{(10^m - 1) - N\}+1$$

$$= \{ \underbrace{999 .....9}_{m \text{ সংখ্যক}} - N \} + 1$$

কোন সংখ্যার নিধানপূরক সংখ্যা নির্ণয়ের সহজ উপায় হল—

প্রথম ধাপ ঃ সংখ্যায় বর্তমান প্রতিটি অঙ্ককে 9 থেকে বিয়োগ কর

দ্বিতীয় ধাপ ঃ প্রথম ধাপে প্রাপ্ত সংখ্যাটির সাথে 1 যোগ কর।

যেমন, $(10^4 - 1547)$ = (9999 – 1547) + 1 = 8452+1= 8453

একই নিয়মে কোন দ্বিপ্রতীকী সংখ্যার নিধানপূরক সংখ্যা বের করা হয়। উদাহরণস্বরূপ বলা যায়, 10110 সংখ্যাটির নিধানপূরক সংখ্যা হল (11111–10110) + 1 = 1001+1 = 1010 অর্থাৎ, প্রদত্ত দ্বিপ্রতীকী সংখ্যাটির

মধ্যে '1' এর পরিবর্তে '0' এবং '0' এর পরিবর্তে '1' বসিয়ে প্রাপ্ত সংখ্যাটির সাথে 1 যোগ করলে প্রদত্ত সংখ্যাটির নিধানপূরক সংখ্যা পাওয়া যায়।

দ্বিপ্রতীকী সংখ্যাপ্রণালীর সংখ্যাগুলির নিধানপূরক সংখ্যাকে 2-পূরক সংখ্যাও বলা হয়।

(খ) **ঊননিধানপূরক সংখ্যাতন্ত্র ঃ** 'ঊন' শব্দের অর্থ-এক কম। যেমন—ঊনপঞ্চাশ, অর্থাৎ, পঞ্চাশ অপেক্ষা এক কম। দশমিক সংখ্যাপ্রণালীতে m সংখ্যক অঙ্কবিশিষ্ট কোন সংখ্যার (N) ঊননিধানপূরক সংখ্যাটি হল

$$\{(10^m - 1) - N\} = \{\underbrace{999...9}_{m \text{ সংখ্যক}} - N\}$$

অর্থাৎ, সংখ্যায় উপস্থিত প্রতিটি অঙ্ককে 9 থেকে বিয়োগ করলে প্রাপ্ত সংখ্যাটি হল প্রদত্ত সংখ্যার ঊননিধান পূরক সংখ্যা। যেমন—1547 সংখ্যাটির ঊননিধান পূরক সংখ্যাটি হল (9999–1547)= 8452।

একই নিয়মে কোন দ্বিপ্রতীকী সংখ্যার ঊননিধান পূরক সংখ্যাটি পেতে হলে সংখ্যাটিতে উপস্থিত প্রতিটি দ্বিচলাঙ্ককে 1-থেকে বিয়োগ করতে হবে। যেমন—101011 সংখ্যাটির ঊননিধান পূরক সংখ্যাটি হল (111111–101011) = 010100। দ্বিপ্রতীকী সংখ্যাপ্রণালীর সংখ্যাগুলির ঊননিধান পূরক সংখ্যাগুলিকে বলা হয় 1-পূরক সংখ্যা।

উদাহরণ ঃ নীচের দ্বিপ্রতীকী সংখ্যাগুলির 1-পূরক সংখ্যা এবং 2-পূরক সংখ্যা বের কর।

(ক) 1 1 0 0 11 (খ) 1 0 1 0 1 0

সমাধান ঃ

(ক)

	1 1 1 1 1 1	
	– 1 1 0 0 1 1	← প্রদত্ত সংখ্যা
	0 0 1 1 0 0	← 1-পূরক সংখ্যা
	+ 1	
	1 1 0 1	← 2- পূরক সংখ্যা

(খ)

	1 1 1 1 1 1	
	– 1 0 1 0 1 0	← প্রদত্ত সংখ্যা
	0 1 0 1 0 1	← 1-পূরক সংখ্যা
	+ 1	
	1 0 1 1 0	← 2- পূরক সংখ্যা

পূরক সংখ্যার সাহায্যে বিয়োগফল নির্ণয় ঃ

বিয়োজিত সংখ্যাটির পূরকসংখ্যাকে বিয়োজ্য সংখ্যার সাথে সুনির্দিষ্ট নিয়মে যোগ করে বিয়োগফল নির্ণয় করা যায়। এই ধরনের বিয়োগফল নির্ণয়পদ্ধতি পরিগণকের আভ্যন্তরীণ গণনাকার্যের জন্য বিশেষ সহায়ক। কারণ, এক্ষেত্রে একই তড়িৎবর্তনী ব্যবহার করে দুটি সংখ্যার যোগফল ও বিয়োগফল নির্ণয় করা সম্ভব হয়। যেহেতু, পূরকসংখ্যা দু'রকমভাবে সংজ্ঞায়িত, পূরকসংখ্যার সাহায্যে বিয়োগফল নির্ণয়পদ্ধতিও দুরকম।

প্রথমে দশমিক সংখ্যাপ্রণালীর জন্য পূরকসংখ্যার সাহায্যে বিয়োগফল নির্ণয়পদ্ধতি আলোচনা করা হল।

(ক) **নিধানপূরক সংখ্যার সাহায্যে বিয়োগ ঃ**

প্রথমে সমসংখ্যক অঙ্কবিশিষ্ট দুটি সংখ্যার বিয়োগফল নির্ণয় পদ্ধতি বলা যাক। যদি বিয়োজিত সংখ্যাটিতে অঙ্কসংখ্যা কম থাকে তবে সংখ্যাটির বামদিকে প্রয়োজনীয় সংখ্যক 0 বসিয়ে প্রদত্ত সংখ্যাদুটিকে সমসংখ্যক করে নিতে হবে। এরপর বিয়োজিত সংখ্যাটির নিধানপূরক সংখ্যা বের করতে হবে। প্রাপ্ত পূরক সংখ্যাটিকে এবার বিয়োজ্য সংখ্যার সাথে যোগ করতে হবে। যোগফলের সর্বাধিক সার্থক অঙ্কটি যদি বাহিত অঙ্ক হিসাবে আসে তাহলে প্রদত্ত সংখ্যাদুটির নির্ণেয় বিয়োগফল হবে ধনাত্মক। এই বাহিত অঙ্কটিকে বাদ দিলে যে সংখ্যাটি পাওয়া যায় সেটিই হল নির্ণেয় বিয়োগ ফল। অন্যদিকে, বিয়োজিত সংখ্যাটির পূরক সংখ্যাকে বিয়োজ্য সংখ্যার সাথে যোগ করার পর প্রাপ্ত যোগফলটির সর্বাধিক সার্থক অঙ্কটি যদি বাহিত অঙ্ক না হয় তাহলে নির্ণেয় বিয়োগফল হবে ঋণাত্মক সংখ্যা। প্রাপ্ত যোগফলটির নিধানপূরক সংখ্যাটি হবে নির্ণেয় বিয়োগফলের সাংখ্যমান।

উদাহরণ ঃ নিধানপূরক সংখ্যার সাহায্যে বিয়োগ কর—

(ক) 82 – 24 (খ) 291 – 132

(গ) 59 – 4 (ঘ) 25 – 48

(ক)
```
  8 2        8 2
 –2 4  →   + 7 6      (∵ 24 এর নিধানপূরক সংখ্যা 76)
          ------
          [1]5 8
        বাহিত অঙ্ক
```

বাহিত অঙ্কটি নির্দেশ করে যে বিয়োগফল ধনাত্মক। ∴ বাহিত অঙ্কটিকে বাদ দিয়ে পাই,

নির্ণেয় বিয়োগফল = + 58

(খ)
```
  2 9 1          2 9 1
 –1 3 2   →    + 8 6 8   (  1 3 2 এর নিধানপূরক সংখ্যা 8 6 8)
               --------
               [1]5 9
        বাহিত অঙ্ক
```

∴ নির্ণয় বিয়োগফল = + 159।

(গ)
```
  5 9        5 9          5 9
  – 4   =  – 0 4   →    + 9 6    (  04 এর নিধানপূরক সংখ্যা 96)
          ---------
           [1]5 5
        বাহিত অঙ্ক
```

∴ নির্ণেয় বিয়োগফল = + 55

(ঘ)
```
  2 5          2 5
 – 4 8   →   + 5 2   (  4 8 এর নিধানপূরক সংখ্যা 5 2)
            -------
               7 7
```

বাহিত অঙ্কের অনুপস্থিতি নির্দেশ করে যে, বিয়োগফল ঋণাত্মক 77 এর নিধানপূরক সংখ্যা হল 23।

অতএব, নির্ণেয় বিয়োগফল = –23

ঊননিধান পূরক সংখ্যার সাহায্যে বিয়োগফল নির্ণয়পদ্ধতি নীচে দেওয়া হল সংক্ষিপ্তাকারে।

1ম ধাপ ঃ যদি বিয়োজিত সংখ্যাটিতে অঙ্কসংখ্যা বিয়োজ্যসংখ্যার অঙ্কসংখ্যা অপেক্ষা কম হয় তাহলে বিয়োজিত সংখ্যাটির বামদিকে প্রয়োজনীয় সংখ্যক শূন্য বসিয়ে নাও, যাতে উভয় সংখ্যাতে সমসংখ্যক অঙ্ক থাকে।

2য় ধাপ ঃ বিয়োজিত সংখ্যাটিতে ঊননিধান পূরক সংখ্যা বের কর এবং সেটিকে বিয়োজ্য সংখ্যার সাথে যোগ কর। যোগফলটির সর্বাধিক সার্থক অঙ্ক বাহিত অঙ্ক হিসাবে আসতে পারে। যদি আসে, তাহলে 3য় ধাপ অণুসরণ কর। অন্যথায় 4থ ধাপ অণুসরণ কর।

3য় ধাপ ঃ যোগফলের সর্বশেষ বাহিত অঙ্কটি অপসারিত কর এবং অবশিষ্ট সংখ্যাটির অসার্থকতম অঙ্কের সাথে 1 যোগ কর। এই রূপে প্রাপ্ত সংখ্যাটি প্রদত্ত সংখ্যাদুটির নির্ণেয় বিয়োগফল, যেটি ধনাত্মক।

4থ ধাপ ঃ এক্ষেত্রে বিয়োগফলটি ঋণাত্মক। প্রাপ্ত যোগফলটির ঊননিধান পূরক সংখ্যাটি বের কর। এটি হল নির্ণেয় বিয়োগফলের সংখ্যামান।

উদাহরণ ঃ ঊননিধান পূরক সংখ্যার সাহায্যে বিয়োগ কর ঃ

(ক) 91 – 45 (খ) 275 – 28 (গ) 34 – 71

সমাধান ঃ

(ক)
```
   91          91
 — 45   →   +  54        (∵ 45 সংখ্যাটির ঊননিধান পূরক সংখ্যা হল 54)
           ---------
            [1] 45
           বাহিত অঙ্ক
```

বাহিত অঙ্ক নির্দেশ করছে যে বিয়োগফল ঋণাত্মক

বাহিত অঙ্কটি অপসারিত করে পাই 45

এখন,
```
   45
 +  1
 -----
  [46]
```

∴ নির্ণেয় বিয়োগফল = + 46

(খ)
```
   275            275        275
 –  28    →   –  028   →  +  971    (∴ 028 এর ঊননিধানপূরক সংখ্যা হল 971)
                          ---------
                           [1] 246
                           বাহিত অঙ্ক
```

বাহিত অঙ্কটি নির্দেশ করছে যে বিয়োগফল ধনাত্মক।

নির্ণেয় বিয়োগফল = + (246 + 1) = + 247

(গ) 34 − 71 → 34 + 28 = 62 (∴ 71 এর ঊননিধানপূরক সংখ্যা হল 28)

$$\begin{array}{r} 34 \\ -\ 71 \\ \hline \end{array} \quad \rightarrow \quad \begin{array}{r} 34 \\ +\ 28 \\ \hline 62 \end{array}$$

যোগফলটির সবশেষে কোন বাহিত অঙ্ক নেই। সুতরাং, বিয়োগফল ঋণাত্মক। যোগফল 62 এর ঊননিধানপূরক সংখ্যা হল 37। অতএব, নির্ণেয় বিয়োগফল = –37.

দ্বিপ্রতীকী সংখ্যাপ্রণালীতে পূরক সংখ্যার সাহায্যে বিয়োগ ঃ

দশমিক সংখ্যাপ্রণালীর মত একই নিয়মে দ্বিপ্রতীকী সংখ্যাপ্রণালীতে পূরক সংখ্যার সাহায্যে বিয়োগ করা যায়। প্রথমে 2-পূরক সংখ্যার সাহায্যে বিয়োগফল নির্ণয়ের পদ্ধতি দেওয়া হল—

1ম ধাপ ঃ বিয়োজিত সংখ্যাটিতে মোট অঙ্কসংখ্যা বিয়োজ্য সংখ্যার অঙ্ক সংখ্যা অপেক্ষা কম হলে বিয়োজিত সংখ্যাটির বামদিক প্রয়োজনীয় সংখ্যাক 0 বসিয়ে নাও যাতে উভয় সংখ্যাতে সমসংখ্যাক অঙ্ক থাকে।

2ম ধাপ ঃ বিয়োজিত সংখ্যাটি 2-পূরক সংখ্যা বের কর। এই পূরকসংখ্যাটি বিয়োজ্য সংখ্যার সাথে যোগ কর। যোগফলটির সর্বাধিক সার্থক অঙ্কটি বাহিত অঙ্ক হিসাবে আসতে পারে অথবা নাও আসতে পারে। যদি আসে, তাহলে 3য় ধাপ অণুসরণ কর। নতুবা, 4থ ধাপ অণুসরণ কর।

3য় ধাপ ঃ এক্ষেত্রে নির্ণেয় বিয়োগফল হবে ধনাত্মক। প্রাপ্ত যোগফলটির সর্বশেষ বাহিত অঙ্কটি অপসারিত করলে যে সংখ্যাটি পাওয়া যাবে সেইটিই হল নির্ণেয় বিয়োগফল।

4র্থ ধাপ ঃ এক্ষেত্রে বিয়োগফলটি হবে ঋণাত্মক। প্রাপ্ত যোগফলটির 2-পূরক সংখ্যা বের কর। এই পূরক সংখ্যাটি হল নির্ণেয় বিয়োগফলের সাংখ্যমান।

উদাহরণ ঃ 2 পূরক সংখ্যার সাহায্যে বিয়োগ করো—

(ক) 1 1 0 1 0 1 0 – 1 0 1 1 0 (খ) 1 0 1 0 1 1 · 01 – 1 1 0 1 · 1

সমাধান ঃ

(ক)

$$\begin{array}{r} 1\ 1\ 0\ 1\ 0\ 1\ 0 \\ -\ 1\ 0\ 1\ 1\ 0 \\ \hline 1\ 1\ 0\ 1\ 0\ 1\ 0 \end{array}$$

বা, $\begin{array}{r} 1\ 1\ 0\ 1\ 0\ 1\ 0 \\ -\ 0\ 0\ 1\ 0\ 1\ 1\ 0 \\ \hline \end{array}$ (বামদিকে দুটি 0 বসিয়ে)

বা, $\begin{array}{r} 1\ 1\ 0\ 1\ 0\ 1\ 0 \\ +\ 1\ 1\ 0\ 1\ 0\ 1\ 0 \\ \hline \boxed{1}\ 1\ 0\ 1\ 0\ 1\ 0\ 0 \end{array}$ (2-পূরক সংখ্যা বসিয়ে)

↑ বাহিত অঙ্ক

∴ নির্ণেয় বিয়োগফল = 1 1 0 1 0 1 0 0

(খ) $\begin{array}{r} 101011\cdot01 \\ -\ 1101\cdot1 \\ \hline \end{array}$

বা, $\begin{array}{r} 101011\cdot01 \\ -\ 001101\cdot10 \end{array}$ (বামদিকে দুটি 0 এবং ডানদিকে একটি 0 বসিয়ে)

বা, $\begin{array}{r} 101011\cdot01 \\ +\ 110010\cdot10 \\ \hline \boxed{1}011101\cdot11 \end{array}$ (2-পূরক সংখ্যা বসিয়ে)

↑ বাহিত অঙ্ক

∴ নির্ণেয় বিয়োগফল = + 0 1 1 1 0 1 · 1 1 = + 1 1 1 0 1 · 11

এবার 1-পূরক সংখ্যার সাহায্যে বিয়োগফল নির্ণেয়র পদ্ধতি দেওযা হল—

1ম ধাপ ঃ যদি বিয়োজিত সংখ্যার মোট অঙ্কসংখ্যা বিয়োজ্য সংখ্যার মোট অঙ্কসংখ্যা অপেক্ষা কম হয় তাহলে বিয়োজিত সংখ্যাটিতে প্রয়োজনীয় সংখ্যক 0 বসিয়ে নাও যাতে উভয় সংখ্যাতে সমসংখ্যক অঙ্ক থাকে।

2য় ধাপ ঃ বিয়োজিত সংখ্যাটির 1-পূরক সংখ্যা বের কর এবং পূরক সংখ্যাটি বিয়োজ্য সংখ্যার সাথে যোগ কর। প্রাপ্ত যোগফলটির সর্বাধিক সার্থক অঙ্ক বাহিত অঙ্ক হিসাবে পাওযা গেলে 3য় ধাপে যাও। নয়তো 4র্থ ধাপে যাও।

3য় ধাপ ঃ এক্ষেত্রে নির্ণেয় বিয়োগফল ধনাত্মক। প্রাপ্ত যোগফলের সর্বশেষ বাহিত অঙ্কটিকে অপসারিত কর এবং অবশিষ্ঠ সংখ্যাটির অসার্থকতম অঙ্কের সাথে 1 যোগ কর। এই যোগফলটি হল প্রদত্ত সংখ্যাদুটির নির্ণেয় বিয়োগফল।

4র্থ ধাপ ঃ এক্ষেত্রে নির্ণেয় বিয়োগফল ঋণাত্মক। 2য়ধাপে প্রাপ্ত যোগফলটির 1-পূরক সংখ্যা বের কর। এই পূরক সংখ্যাটি হল নির্ণেয় বিয়োগফলের সংখ্যামান।

উদাহরণ ঃ 1-পূরক সংখ্যার সাহায্যে বিয়োগ কর—

(ক) 111000 – 1011 (খ) 101010·11 – 110·01

সমাধান ঃ

(ক) $\begin{array}{r} 111000 \\ -\ 1011 \\ \hline \end{array}$ বা, $\begin{array}{r} 000111 \\ -\ 001011 \\ \hline \end{array}$ বা, $\begin{array}{r} 111000 \\ +\ 110100 \\ \hline \boxed{1}101100 \end{array}$

বাহিত অঙ্ক

\ সর্বাধিক সার্থক অঙ্কটি বাহিত অঙ্ক, নির্ণেয় বিয়োগফল ধনাত্মক। বাহিত অঙ্কটি অপসারিত করে পাই 1 0 1 1 0 0।

এখন,

```
   1 0 1 1 0 0
 +           1
 -------------
   1 0 1 1 0 1
```

\ নির্ণেয় বিয়োগফল = 1 0 1 1 0 1

(খ)

```
  1 0 1 0 1 0 · 1 1
      – 1 1 0 · 0 1
  -----------------
```

বা,

```
    1 0 1 0 1 0 · 1 1
  – 0 0 0 1 1 0 · 0 1     ( বামদিকে তিনটি 0 বসিয়ে)
  -------------------
```

বা,

```
    1 0 1 0 1 0 · 1 1
  + 1 1 1 0 0 1 · 1 0     ( 1-পূরক সংখ্যা বসিয়ে)
  -------------------
  [1] 1 0 0 1 0 0 · 0 1
   ↑
  বাহিত অঙ্ক
```

বাহিত অঙ্ক অপসারিত করে পাই, 1 0 0 1 0 0 ·0 1

এই সংখ্যাটির সর্বনিম্ন সার্থক অঙ্কের সাথে 1 যোগ করে পাই,

```
  1 0 0 1 0 0 · 0 1
+                 1
-------------------
  1 0 0 1 0 0 · 1 0
```

নির্ণেয় বিয়োগফল = 1 0 0 1 0 0 · 1 0

দ্বিপ্রতীকী সংখ্যাপ্রণালীতে ভাগফল নির্ণয় ঃ দশমিক সংখ্যাপ্রণালীতে যেভাবে ভাগ করা হয় একই ভাবে দ্বিপ্রতীকী সংখ্যাপ্রণালীতেও ভাগ করা যায়। নীচে দুটি দ্বিচলাঙ্কের জন্য ভাগ প্রক্রিয়ার নিয়ম দেওয়া হল।

ভাজ্য	ভাজক	গাণিতিক রাশিমালা	ভাগফল	ভাগশেষ
1	1	1 ¸ 1	1	0
0	1	0 ¸ 1	0	0
0	0	0 ¸ 0	অর্থহীন	
1	0	1 ¸ 0	অর্থহীন	

উদাহরণ ঃ ভাগফল ও ভাগশেষ নির্ণয় কর—

(ক) 1 1 0 1 0 1 1 ¸ 1 0 1 (খ) 1 0 1 1 1 0 · 1 0 1 ¸ 1 0 1

সমাধান ঃ

(ক) ভাজক ভাজ্য ভাগফল

```
101) 1 1 0 1 0 1 1 (10101
     1 0 1
   ___________
        1 1 0
        1 0 1
        _________
           1 1 1
           1 0 1
           _______
             1 0   ← ভাগশেষ
```

ভাজক ভাজ্য ভাগফল

```
110 ) 10 111 0· 101 ( 111 · 11 0001
      1 10
  _____________
     1011
      110
  _____________
      1010
       110
  _____________
       1001
        110
  _____________
        110
        110
  _____________
           1001
            110
           ______
             11
          ...............
          ...............
```

∴ নির্ণেয় ভাগফল = 111 · 110 (অর্ধাংশবিন্দুর পর দুই অঙ্ক পর্যন্ত শুদ্ধ)

দ্বিপ্রতীকী সংকেতলিপির সাহায্যে দশমিক সংখ্যার প্রকাশ (Binary coded decimal number representation/BCD number representation) ঃ

কোন দশমিক সংখ্যায় বর্তমান প্রতিটি অঙ্ককে সমমান দ্বিপ্রতীকী সংখ্যা দ্বারা প্রতিস্থাপিত করবার যে প্রক্রিয়া

সেটি গোড়ার দিকের পরিগণকগুলিতে যথেষ্ট ব্যবহার করা হত। 0, 1, 2, ..., 8, 9 এই দশটি এক অঙ্কবিশিষ্ট দশমিক সংখ্যাকে চার অঙ্কবিশিষ্ট দ্বিপ্রতীকী সংখ্যার সাহায্যে প্রতিস্থাপিত করা সম্ভব। এই প্রক্রিয়া অনুযায়ী দশমিক সংখ্যা 138 কে লিখতে হবে

$(138)_{10}$

$= (\ \underbrace{1}\quad \underbrace{3}\quad \underbrace{8}\ )_{10}$

$= (0001\quad 0011\quad 1000\ )_{BCD}$

এই প্রক্রিয়ায় সংখ্যাটির চিহ্ন বোঝাতে সংখ্যাটির পূর্বে আরও চারটি দ্বিচলাঙ্কের সমাহার ব্যবহার করতে হবে। '+' চিহ্ন বোঝাতে 1010 এবং '–' চিহ্ন বোঝাতে 1011 যোগ করা যেতে পারে। অতএব, পাঁচ অঙ্কবিশিষ্ট কোন দশমিক সংখ্যাকে চিহ্নসমেত বোঝাতে হলে 5×4+4=24 টি অঙ্কবিশিষ্ট একটি দ্বিপ্রতীকী উপস্থাপনার সাহায্য নিতে হবে। এইরূপ দ্বিপ্রতীকী সংকেতলিপির সাহায্যে দশমিক সংখ্যার প্রকাশ পদ্ধতিটি ইংরাজীতে BCD number representation নামে প্রচলিত।

তথ্যের প্রকৃতি ও তথ্য প্রকাশনা ঃ

এক কথায় তথ্যের সংজ্ঞা দেওয়া সম্ভব নয়। কিন্তু একটি উদাহরণের মাধ্যমে সহজেই বুঝে নেওয়া সম্ভব—তথ্য কি? যেমন ধরা যাক, পরীক্ষার ফল প্রকাশের দিন সংশ্লিষ্ট দপ্তর থেকে জানানো হল যে—150 জন পরীক্ষার্থীর মধ্যে 145 জন কৃতকার্য হয়েছে ; সর্বোচ্চ স্থান অধিকার করেছে অমৃতা এবং দ্বিতীয় স্থানাধিকারী আমন।

সংশ্লিষ্ট দপ্তর থেকে প্রকাশিত বা ঘোষিত উপরের বার্তাটিতে (information) যে বিশেষ দুটি সংখ্যা—150 ও 146, এবং যে দুটি সুনির্দিষ্ট নামের—অমৃতা ও আমন, এর উল্লেখ রয়েছে সেগুলি হল এই বার্তার সাথে সংশ্লিষ্ট তথ্য। এই উদাহরণটি থেকে আরও বোঝা যাচ্ছে যে, প্রকৃতিগতভাবে তথ্য দু'ধরণের—সাংখ্যিক এবং অসাংখ্যিক (বা বর্ণমালা প্রকাশিত)। উপরের উদাহরণটিতে সাংখ্যিক তথ্য হল—150 ও 146 এবং অসাংখ্যকি তথ্য হল—অমৃতা ও আমন। এছাড়া আমরা অনেক ক্ষেত্রে মিশ্রপ্রকৃতির তথ্যও দেখতে পাই। যেমন, কোন পরিগণকের জন্য তার কোন ব্যবহারকারীর বৈধতাসূচক (বা অনুমতিদায়ক) সংকেতশব্দ (ইংরাজীতে password) হতে পারে 'apw1403'. প্রকৃতিগতভাবে 'apw1403' সঙ্কেতশব্দটি মিশ্র প্রকৃতির তথ্য বা সাংখ্য-বর্ণমালা প্রকাশিত।

প্রথম উদাহরণটিতে, পরীক্ষার ফলাফল সম্পর্কিত যে নির্দেশ বা ঘোষণার উল্লেখ রয়েছে সেটি একটি বার্তা বা সংবাদ (information)। আবার এই ঘোষণা থেকে আমরা পরীক্ষায় শতকরা কৃতকার্যের হার, শতকরা অকৃতকার্যের হার ইত্যাদি জ্ঞানলাভ করতে পারি। এই নতুন জ্ঞান হল আমাদের কাছে নতুন বার্তা (information)। আবার সমস্ত পরীক্ষার্থীর জন্য মেধাতালিকাও তৈরী করা সম্ভব। এই মেধাতালিকাটি হবে আরও একটি নতুন বার্তা। বর্তমানে যে সমস্ত পরিগণক আমরা ব্যবহার করি সেগুলি সাংখ্যিক, অসাংখ্যিক ও মিশ্র প্রকৃতির তথ্য সঞ্চয় করতে সক্ষম এবং সেই সমস্ত তথ্য বিশ্লেষণ করে আমাদের কাছে নতুন বার্তাও পৌঁছে দিতে পারে। যেমন, শতকরা কৃতকার্যের হার বার করতে প্রয়োজনীয় গাণিতিক হিসাব করা সাংখ্যিক তথ্য বিশ্লেষণ) বা মেধাভিত্তিক নামের তালিকা প্রস্তুতি (সাংখ্যিক ও অসাংখ্যিক তথ্য বিশ্লেষণ)—এই দুই'ই পরিগণকের পক্ষে সহজসাধ্য কাজ।

বিট, বাইট ও শব্দ (BIT, BYTE and WORD) ঃ

কোনও তথ্যকে লিখিতভাবে প্রকাশ করতে আমরা ব্যবহার করি

(i) কোন সংখ্যা যা, 0, 1, ..., 9 পর্যন্ত অঙ্ক দ্বারা প্রকাশিত (মোট 10 টি চিহ্ন)

(ii) বর্ণমালার বিভিন্ন বর্ণ, যেমন ইংরেজী বর্ণমালার A, B, ..., Z এবং a, b,...., z (মোট 26+26 =52 টি চিহ্ন)

(iii) বিশেষ চিহ্ন, যেমন +, –, = , ?, &, #, <, ইত্যাদি। (প্রায় 32 টি চিহ্ন)

তথ্য প্রকাশনায় ব্যবহৃত এই সমস্ত ক্ষুদ্রতম চিহ্ন বা অঙ্ক বা বর্ণকে পরিগণক বিজ্ঞানে সাধারণভাবে বলা হয় (তথ্যকোষাঙ্ক) বা ইংরাজীতে character। উপরে উল্লিখিত তথ্যকোষাঙ্কগুলি ছাড়া আরও কয়েকটি নিয়ন্ত্রক চিহ্নও ব্যবহার করা হয়। এদের মধ্যে কয়েকটি পরিগণকে চিত্রাঙ্কাণের কাজে ব্যবহৃত হয়।

বিট (BIT) ঃ

বিট শব্দটি প্রকৃতপক্ষে BInary digiT এর সংক্ষিপ্ত আকার। 0 এবং 1 এই দুটি দ্বিপ্রতীকী সংখ্যা বা দ্বিচলাঙ্ককে এক কথায় বলা হয় বিট। দ্বিপ্রতীকী সংখ্যাপ্রণালীতে যে কোন সংখ্যার লিখিত প্রকাশ হল কয়েকটি বিট-এর সমাহার।

বাইট (Byte) ঃ

বাইট শব্দটি এসেছে 'by eight' এর সংক্ষিপ্ত রূপ হিসাবে। পরিগণকের আভ্যন্তরীণ ক্রিয়াসম্পাদন, যেমন—তথ্য সঞ্চয়, তথ্য বিশ্লেষণ, তথ্যের আদান-প্রদান,—এই সবই হয় বিট এর সুনির্দিষ্ট ব্যবহার দ্বারা। একটি তথ্যকে অন্য তথ্যের থেকে আলাদাভাবে বুঝতে, যেমন, একটি সংখ্যাকে অন্যান্যসংখ্যা থেকে আলাদাভাবে বুঝতে, একটি বর্ণকে অন্যান্য বর্ণ থেকে পৃথকীকরণের জন্য সুনির্দিষ্ট সঙ্কেতলিপির প্রয়োজন। এই সঙ্কেতলিপি তৈরী করা যেতে পারে কয়েকটি বিট এর সমাহার দ্বারা। এই ধারণা প্রথম পাওয়া যায় জন ভন নয়ম্যান (John Von neumann) এর কাছে 1946 সালে। তথ্যের লিখিত প্রকাশের জন্য প্রয়োজনীয় সমস্ত character গুলিকে বা তথ্যকোষাঙ্কগুলিকে কয়েকটি বিট-এর সমাহার দ্বারা সুনির্দিষ্টভাবে প্রকাশ করে বিভিন্ন সঙ্কেতলিপি তৈরী করা যায়। বর্তমানে যে সমস্ত সংকেতলিপি ব্যবহার করা হয় সেগুলির মধ্যে বিশেষ উল্লেখযোগ্য হল ASCII সংকেতলিপি, EBCDIC সংকেতলিপি। এই সংকেতলিপি দুটির প্রত্যেকটি হল দ্বিপ্রতীকী সংকেতলিপি এবং এক একটি charater—কে বোঝাতে বর্তমানে প্রতিটি সংকেতলিপিতে আটটি দ্বিচলাঙ্ক বিশিষ্ট সংকেত ব্যবহৃত হয়েছে। আটটি দ্বিচলাঙ্কের সমাহার বা আটটি বিটের সমাহারে তৈরী এক একটি সঙ্কেতকে বলা হয় একটি বাইট। এক একটি বাইট নির্দেশ করে এক একটি তথ্যকোষাঙ্ককে।

শব্দ (Word) ঃ

পরিগণের একটি শব্দ বা Word তৈরী হয় কয়েকটি বাইটের সমাহার দ্বারা। বর্তমানে দুটি, চারটি বা আটটি বাইটের সমাহারবিশিষ্ট পরিগণকীয় শব্দ ব্যবহৃত হয়।

পরিগণকের স্মৃতিকক্ষে কোন শব্দকে সঞ্চয় করবার সময় একটি বিশেষ আভ্যন্তরীণ নিয়মমাফিক একটি সুনির্দিষ্ট সংখ্যাও নির্ধারিত হয়ে যায়। এই সুনির্দিষ্ট সংখ্যাটি হল ঐ পরিগণকীয় শব্দের ঠিকানা বা address। সুতরাং, যদি কোন পরিগণকের স্মৃতিকক্ষের শব্দসঞ্চয়সীমা 2^{16} বা 65536 সংখ্যক শব্দ বলে ধরা হয় তবে এই বিভিন্ন শব্দগুলির ঠিকানা সংখ্যা হবে 0, 1, 2, 3, ..., 65535, 65536।

সংকেতলিপি (ASCII) ঃ

ASCII শব্দটির পূর্ণরূপ হল American Standard Code for Information Interchange. ASCII সংকেতলিপি হল একটি দ্বিপ্রতীকী সঙ্কেতলিপি। বর্তমানে এই সঙ্কেতলিপির ব্যবহার সর্বাধিক। গোড়ার দিকে এই সঙ্কেতলিপিতে ছয়টি দ্বিচলাঙ্ক বিশিষ্ট বা ছয়টি বিট-সমন্বিত সংকেত ব্যবহার করা হত। ক্রমশঃ সময়ের অগ্রগতির সাথে সাথে পরিগণকে, কর্ম সম্পাদনে ব্যবহৃত উচ্চমানের ভাষা প্রকৃতই উন্নতমানের হতে থাকে। বর্তমানে বড় হরফের A, B, C, D, ...ইত্যাদি বর্ণকে ছোট হরফের a, b, c, d, ... ইত্যাদি বর্ণ থেকে পৃথকভাবে বুঝতে হবে। অর্থাৎ পরিগণকের ভাষাকে প্রাঞ্জল করার জন্য, ভাষাকে সাধারণের পক্ষে সহজবোধ্য করে তুলতে পূর্বের তুলনায় অনেক বেশি তথ্যকোষাঙ্ক এবং সেইসঙ্গে অনেক বেশি সঙ্কেত ব্যবহার করতে হয়। সুতরাং, ছয়টি বিটের সমাহার দ্বারা তৈরী সংকেতলিপির সাহায্যে প্রয়োজনীয় সংখ্যক তথ্যকোষাঙ্ককে লিপিবদ্ধ করা সম্ভব নয়। তাই বর্তমানে প্রচলিত ASCII সংকেতলিপিতে আটটি বিটের সমাহারে (অর্থাৎ এক বাইট) তৈরী সংকেত দ্বারা এক একটি তথ্যকোষাঙ্ককে সূচিত করা হয়। যেহেতু একটি বিট হতে পারে 0 বা 1 এবং আট অঙ্কের সঙ্কেতে যে কোন অঙ্ক হতে পারে 0 বা 1, সেহেতু, আটটি বিটের সমাহার দ্বারা 2^8 = 256 টি বিভিন্ন সংকেত পাওয়া যায়। এই সমস্ত সংকেত দ্বারা নির্দেশিত তথ্যকোষাঙ্কগুলিকে নিম্নলিখিত ভাবে বিভিন্ন শ্রেণীতে বিভাজন করা যেতে পারে—

(i) বড় হরফের ইংরেজী বর্ণ, যেমন A, B, C, D, ... (মোট 26 টি)

(ii) ছোট হরফের ইংরেজী বর্ণ, যেমন a, b, c, d, ... (মোট 26 টি)

(iii) 0 থেকে 9 পর্যন্ত দশমিক সংখ্যা (মোট 10 টি)

(iv) বিশেষ চিহ্ন, যেমন +, –, (,), ?, # ..... (মোট 32 টি)

(v) নিয়ন্ত্রক চিহ্ন (মোট 34 টি)

(vi) লেখচিত্র সহায়ক চিহ্ন (/চিত্রাঙ্কণ সহায়ক চিহ্ন) (মোট 128 টি)

নীচের টেবিলিতে ASCII সংকেতলিপিতে ব্যবহৃত কিছু সংকেতকে দেখানো হল।

8 বিট ASCII সঙ্কেত

মূখ্য সার্থক অঙ্ক b_7 b_6 b_5 b_4								গৌণ সার্থক অঙ্ক			
0000	0001	0010	0011	0100	0101	0110	0111	b_3	b_2	b_1	b_0
NUL	DLE	SPACE	0	@	P		p	0	0	0	0
SOH	DC 1	!	1	A	Q	a	q	0	0	0	1
STX	DC 2	,,	2	B	R	b	r	0	0	1	0
ETX	DC 3	#	3	C	S	c	s	0	0	1	1
EOT	DC 4	$	4	D	T	d	t	0	1	0	0
ENQ	NAK	%	5	E	U	e	u	0	1	0	1
ACK	SYN	&	6	F	V	f	v	0	1	1	0
BEL	ETB		7	G	W	g	w	0	1	1	1
BS	CAN	(	8	H	X	h	x	1	0	0	0
HT	EM	)	9	I	Y	i	y	1	0	0	1
LF	SUB	*	:	J	Z	j	z	1	0	1	0
VT	ESC	+	;	K	[	k	{	1	0	1	1
FF	FS	’	<	L	\	l	l	1	1	0	0
CR	GS	–	=	M	]	m	}	1	1	0	1
SO	RS	·	>	N	^	n	~	1	1	1	0
SI	US	/	?	O	–	0	DEL	1	1	1	1

ASCII সঙ্কেতলিপি অনুযায়ী Maths. এর সংকেত লিপি হবে

এখানে ASCIICode এর টেবিলে যে সমস্ত তথ্যকোষাঙ্ক এবং নিয়ন্ত্রক নির্দেশের দ্বিপ্রতীকী সংকেত লিপি দেওয়া হয়েছে সেগুলি ছাড়া আরও বেশ কিছু সংকেতলিপির উল্লেখ রয়েছে মূল ASCII সংকেতলিপিতে। যেমন—α, β, $\propto$, ε, $\surd$, End of line, End of page ইত্যাদি।

E B C D I C সংকেতলিপি ঃ

A S C I I সংকেতলিপির যত EBCDIC সংকেতলিপিতেও 8-বিট সংকেত ব্যবহার করা হয়। EBCDIC সংকেতলিপিতে ব্যবহৃত সংকেত গুলি ASCII সংকেত-এর মতো ; কিন্তু মূখ্য সার্থক অঙ্কগুলি আলাদা। EBCDIC সংকেতলিপির ব্যবহার হয় মূলত IBM (International Business Machines) কর্তৃক উৎপাদিত পরিগণক যন্ত্রে। কোন তথ্য যদি ASCII সংকেত অনুযায়ী করা থাকে এবং সেই তথ্য যদি কেউ এমন একটি পরিগণক যন্ত্রে কাজে লাগাতে চায় যেটি EDCDIC কোড ব্যবহার করে, তবে এই রূপান্তরকার্য করা যেতে পারে প্রোগ্রাম করে।

সংক্ষিপ্তসার ঃ

এই পাঠে প্রধানত বিভিন্ন প্রকারের সংখ্যাপ্রণালী যথা—দ্বিপ্রতীকী সংখ্যা প্রণালী, অষ্টপ্রতীকী সংখ্যা প্রণালী, ষোড়শ প্রতীকী সংখ্যা প্রণালী ও দশমিক সংখ্যা প্রণালী বিশদ আলোচনা করা হয়েছে। কোন একটি সংখ্যা প্রণালীর সংখ্যা অন্য কোন সংখ্যা প্রণালীতে কিভাবে পরিবর্তন করা হয় তা উদাহরণ সহযোগে বর্ণনা করা হয়েছে। বিভিন্ন সংখ্যা প্রণালীর সংখ্যার যোগ, বিয়োগ, গুন ও ভাগ উদাহরণ সহ বর্ণনা করা হয়েছে। নিধানপূরক সংখ্যাতন্ত্র কি ও বিয়োগের ক্ষেত্রে ইহার উপযোগিতা কি তাও উদাহরণ সহ আলোচনা করা হয়েছে। এই পাঠে দ্বিপ্রতীকী সংকেত লিপির সাহায্যে দশমিক সংখ্যার প্রকাশ পদ্ধতি বর্ণনা করা হয়েছে। তথ্যের প্রকৃতি ও তথ্য প্রকাশনা সম্বন্ধে সংক্ষিপ্ত বর্ণনা পাঠের শেষে দেওয়া আছে।

একক 3 ❑ নির্দেশলিপিকরণ ভাষা (Programming Language)

এককের উদ্দেশ্য ঃ

এই একক পাঠ করলে পাঠক নিম্নলিখিত বিষয়গুলি সম্বন্ধে জ্ঞান লাভ করবে।

- পরিগণক যন্ত্রের ভাষা
- পরিগণক যন্ত্রের নিম্নমানের ও উচ্চমানের ভাষা
- সংযোজনী ভাষা
- উচ্চমানের ভাষা
- প্রোগ্রাম কি?
- উৎস ও যান্ত্রিক প্রোগ্রাম সমূহ
- সংকলক এবং অনুবাদক
- বহুল ব্যবহৃত কয়েকটি উচ্চমানের ভাষা—BASIC, FORTRAN, COBOL.

নির্দেশলিপিকরণ ভাষা

ভাষা হল দুটি মানুষের মধ্যে ভাববিনিময়ের মাধ্যম। যেমন—বাংলা, ইংরাজী, হিন্দি, আরবী, পার্সী, রুশ ইত্যাদি। ভাষার মাধ্যমে একটি মানুষ আর একটি মানুষের কাছে নিজের মনোভাব ব্যক্ত করে এবং এইভাবে নিজের কাজ চালায়। কিন্তু পরিগণকের যেহেতু নিজস্ব বোধশক্তি নেই, প্রকৃত বুদ্ধিস্বত্ত্বা নেই এবং কোন ভাবাবেগও নেই তাই মানুষের দৈনন্দিন জীবনে ব্যবহৃত ভাষা তার কাছে সম্পূর্ণ অর্থহীন। তাহলে একটি পরিগণককে মানুষ কিভাবে কাজে লাগাবে? কিভাবে একটি পরিগণককে মানুষ কর্ম সম্পাদনের নির্দেশ দেবে? মানুষের সাথে পরিগণকের কর্মযোগসূত্র তৈরী করতে চাই বিশেষ কোন ভাষা মাধ্যম, যা সাধারণ কথোপকথনের ভাষা থেকে ভিন্ন চরিত্রের। বলা বাহুল্য যে, এই ভাষায় মানুষ একটি পরিগণককে মূলতঃ কিছু কর্মসম্পাদনের নির্দেশই দেবে। অর্থাৎ, ভাষাটি হবে নির্দেশমূলক এবং সেই নির্দেশানুসারে পরিগণক কাজ করবে। সুতরাং এই ভাষাকে যদি বলা যায় পরিগণকের কর্মভাষা তাহলে বাংলা, হিন্দি, ইংরাজী ইত্যাদি হল মানুষের কর্মভাষা। পরিগণককে কর্ম নির্দেশ দিতে পারে এমন ভাষাকে বলা হবে পরিগণকের কর্মভাষা (Computer Language) বা নির্দেশলিপিকরণ ভাষা (Programming Language)।

পরিগণকের কর্মপদ্ধতি উদ্ভাবনের সময় থেকে শুরু করে আজ পর্যন্ত বিভিন্ন নির্দেশলিপিকরণ ভাষা উদ্ভাবিত হয়েছে। পরিগণকের কর্মভাষার মান নির্ভর করে দুটি বিষয়ের উপর—প্রথমতঃ ভাষাটি কতটা সরল ; দ্বিতীয়তঃ ঐ ভাষায় কর্মনির্দেশ লিপি তৈরী করা কতটা সহজ। স্বভাবতঃই পরিগণকের যে কর্মভাষায় যত বেশী মানুষের কর্মভাষায় ব্যবহৃত শব্দাদির উল্লেখ থাকবে সেই ভাষা তত বেশী সাধারণ মানুষের ব্যবহার উপযোগী এবং জনপ্রিয় হবে।

মান অনুযায়ী পরিগণকের নির্দেশলিপিকরণ ভাষা দু'ধরণের—(ক) নিম্নমানের ভাষা (Low Level Language) (খ) উচ্চমাণের ভাষা (High Level Language)।

নিম্নমানের ভাষার উদাহরণ হল—

(i) যান্ত্রিক ভাষা (Machine Language)
(ii) সংযোজনী ভাষা (Assembly Language)

যান্ত্রিক ভাষা :

এই ভাষায় শুধুমাত্র 0 ও 1 এর সমাহারে তৈরী সাংকেতিক নির্দেশ ব্যবহার করে পরিগণকের কর্মনির্দেশলিপি বা ক্রিয়াদেশলিপি (ইংরাজীতে Program) তৈরী করা হয়। যেহেতু দুটি ভিন্ন বিভববিশিষ্ট বৈদ্যুতিক তরঙ্গ ব্যবহার করে 0 এবং 1 কে আলাদাভাবে বোঝানো সম্ভব, সেহেতু যান্ত্রিক ভাষায় লেখা কোন ক্রিয়াদেশলিপি খুব সহজেই পরিগণকের পক্ষে কার্যকরী হতে পারে। কিন্তু যে মানুষটি এই ক্রিয়াদেশলিপি তৈরী করছেন অর্থাৎ ক্রিয়াদেশলিপিকারকের পক্ষে যান্ত্রিক ভাষা মনে রাখা বা এই ভাষা ব্যবহার করে পরিগণকের কর্মনির্দেশলিপি তৈরী করা একেবারেই সহজসাধ্য নয়। অতএব, যান্ত্রিক ভাষায় কাউকে সুদীর্ঘ ক্রিয়াদেশলিপি তৈরী করতে বলা হলে সেটি তার পক্ষে যন্ত্রনাদায়ক হয়ে উঠবে। নীচে যান্ত্রিক ভাষায় ব্যবহৃত কিছু নির্দেশ-এর উল্লেখ করা হল, যা থেকে বোঝা যাবে সাধারণ মানুষের পক্ষে যান্ত্রিক ভাষার ব্যবহার কতটা দুরূহ।

যান্ত্রিক ভাষায় ব্যবহৃত প্রতিটি নির্দেশের দুটি অংশ। প্রথম অংশটিতে থাকে কোন ক্রিয়ানির্দেশ, দ্বিতীয় অংশটিতে থাকে কোন তথ্য বা তথ্যের ঠিকানা। প্রথম অংশটিকে বলা হয় ক্রিয়ানির্দেশ সংকেত

Operation code (সংক্ষেপে OP CODE) / Instruction code/ Function code], দ্বিতীয় অংশটিকে বলা হয় ক্রিয়া স্থান সংকেত (Operand)। যেমন—

STM	205
OPCODE	OPERAND

এটি একটি যান্ত্রিক ভাষায় লিখিত নির্দেশ। এই নির্দেশটির অর্থ হল—সঞ্চায়কের (accumulator) আধেয়টিকে (content) জমা করে রাখ পরিগণকের স্মৃতিকক্ষে বা স্মৃতিআধারে) (Store into Memory) 205 সংখ্যাদ্বারা চিহ্নিত ঠিকানায়।

এবার, 24 এবং 67 সংখ্যাদুটি যোগ করার জন্য যান্ত্রিক ভাষায় লেখা একটি ক্রিয়ানির্দেশলিপি নীচে দেওয়া হল—

1ম নির্দেশ :	LDV	24	LoaD a Value 24 অর্থাৎ 24 সংখ্যাটি সঞ্চায়কে অর্পণ কর
2য় নির্দেশ :	STM	3501	STore into Memory অর্থাৎ সঞ্চায়কে বর্তমান সংখ্যাটি পরিগণকের স্মৃতিকক্ষে 3501 সংখ্যাদ্বারা চিহ্নিত ঠিকানায় গচ্ছিত রাখ
3য় নির্দেশ :	LDV	67	LoaD a Value 67 অর্থাৎ 67 সংখ্যাটি সঞ্চায়ককে অর্পণ কর। এর ফলে পূর্বের 24 সংখ্যাটি অপসারিত হয়ে সঞ্চায়কে 67 সংখ্যাটি আসবে
4র্থ নির্দেশ :	ADD	3501	3501 সংখ্যাদ্বারা সূচিত ঠিকানায় যে সংখ্যাটি জমা আছে

			সেটিকে সঞ্চায়কে বর্তমান সংখ্যাটির সাথে যোগ কর। এই যোগফলটি সঞ্চায়কেই থেকে যাবে
5ম নির্দেশ :	STM	4106	সঞ্চায়কে বর্তমান যোগফলটি স্মৃতিকক্ষে 4106 সংখ্যক ঠিকানাতে জমা করে রাখ

এই নির্দেশলিপিতে যে নির্দেশগুলি দেওয়া হয়েছে সেগুলিতে একটির বেশী ঠিকানা—সংখ্যা ব্যবহার করা হয়নি। পরিগণকের যান্ত্রিক ভাষাতে দুই ঠিকানা সম্বলিত নির্দেশও ব্যবহার করা হয়। এক্ষেত্রে উপরের নির্দেশলিপির পরিবর্তে নীচের নির্দেশলিপিটিও সমান কার্যকর।

LDV	24	
STM	3501	
LDV	67	
STM	3674	
ADD	3501, 3674	[3501 এবং 3674 নং ঠিকানায় গচ্ছিত সংখ্যা
STM	4106	দুটি যোগ কর]

উভয় ক্ষেত্রেই 4106 ঠিকানায় জমা হওয়া সংখ্যাটি হবে 91 (= 24 + 67). এখানে LDV, STM, ADD ইত্যাদি যে সমস্ত ক্রিয়ানির্দেশ ব্যবহার করা হয়েছে এদের প্রত্যেকটির জন্য দ্বিপ্রতীকী নির্দেশসঙ্কেত রয়েছে। এই ক্রিয়ানির্দেশসঙ্কেত গুলি তৈরী করেন কোন পরিগণকের যান্ত্রিক ভাষার রচয়িতা। তিনি এই কাজটি করেন পরিগণকের আভ্যন্তরীণ গঠনে ব্যবহৃত উপাদানের প্রকৃতি অনুসারে। যেমন—তথ্যনিবন্ধক (Registers), সঞ্চায়ক (accumulator), স্মৃতিআধারের ধারণক্ষমতা, পরিগণকীয় শব্দদৈর্ঘ্য (Computer word length) ইত্যাদী বিষয়কে বিবেচনা করে যান্ত্রিক ভাষার সংকেতলিপি তৈরী করতে হয়। সুতরাং, যান্ত্রিক ভাষা বিভিন্ন গঠনের পরিগণকের জন্য বিভিন্ন হবে।

পরিগণকীয় ভাষা বিবর্তনের গোড়ার দিকে সমস্ত নির্দেশলিপি তৈরী করতে হত যান্ত্রিক ভাষায়। খুব ছোট কোন গাণিতিক সমস্যা সমাধানের জন্য তৈরী নির্দেশলিপিটিও যথেষ্ট দীর্ঘ হত। নির্দেশলিপিটি তৈরী করবার সময় ব্যবহৃত প্রতিটি তথ্যের ঠিকানা সংখ্যা মনে রাখতে হত লিপিকারককে। লিপিকরণের সময় কোথাও কোন ভ্রান্তি হলে সেটিকে পরে খুঁজে বার করা ছিল বিশেষ দুরূহ কাজ।

সংযোজনী ভাষা বা প্রতীকী যন্ত্রভাষা (Assembly Language) ঃ

এটিও একটি নিম্নমাণের ভাষা। এই ভাষাটির উদ্ভাবন হয়েছিল 1950 সালের শেষের দিকে। যান্ত্রিক ভাষার ক্রিয়ানির্দেশলিপি তৈরী করার মধ্যে যে প্রতিকূলতা ছিল বা যে অতিসতর্কতার আবশ্যক ছিল সেটিকে কিছুটা লঘু করার জন্য সংযোজনী ভাষার উদ্ভাবন হয়। এই ভাষাতে যে সমস্ত ক্রিয়ানির্দেশসঙ্কেত এবং ঠিকানানির্দেশক সঙ্কেত ব্যবহার করা হয় সেগুলি যান্ত্রিক ভাষার ব্যবহৃত সঙ্কেতাদির তুলনায় অনেক বেশি স্মৃতিসহায়ক। এরকারণ হল এই যে, যান্ত্রিক ভাষায় সমস্ত সঙ্কেতই দ্বিপ্রতীকী অর্থাৎ শুধুমাত্র বিট দ্বারা প্রকাশিত ; কিন্তু সংযোজনী ভাষায় ব্যবহৃত সঙ্কেতগুলিতে ইংরাজী বর্ণ বা অক্ষর, দশমিক সংখ্যা ইত্যাদি ব্যবহার করা হয়েছে। এর ফলে নির্দেশলিপিকারকে সুদীর্ঘ দ্বিপ্রতীকী সঙ্কেতলিপি মনে করে রাখবার প্রয়োজন থাকছে না। যেমন, দুটি সংখ্যার বিয়োগফল নির্ণয় করতে যে ক্রিয়াদেশলিপি দরকার তা সংযোজনী ভাষায় নিম্নরূপ—

1ম নির্দেশ :	READ	FIRST, SECOND	[অর্থাৎ FIRST এবং SECOND এই দুটি চলরাশির মান পরিগণকের স্মৃতিকক্ষে জমা করে রাখ]
2য় নির্দেশ :	LDA	FIRST	[FIRST এই চলরাশির মান সঞ্চায়কে নিয়ে এস]
3য় নির্দেশ :	SUB	SECOND	[SECOND চলরাশির মানকে সঞ্চায়কে বর্তমান সংখ্যা থেকে বিয়োগ কর। বিয়োগফল সঞ্চায়কে থেকে যাবে]

উপরের এই উদাহরণটি থেকে বুঝতে অসুবিধা হয় না যে, নির্দেশলিপিকারকের কাছে যান্ত্রিক ভাষার তুলনায় সংযোজনী ভাষা বেশি অনুকূল। কিন্তু এই ভাষাটিও যন্ত্রনির্ভর। অর্থাৎ পরিগণকের আভ্যন্তরীণ বিন্যাসভেদ অনুযায়ী সংযোজনী ভাষায় ব্যবহৃত ক্রিয়ানির্দেশসঙ্কেতগুলি বিভিন্ন হতে পারে।

পরিগণক যেহেতু শুধুমাত্র '0' এবং '1' কে আলাদাভাবে বুঝতে সক্ষম সেহেতু সংযোজনী ভাষায় লেখা ক্রিয়াদেশলিপিকে যান্ত্রিক ভাষাতে অনুবাদ (translation) করে নেওয়া দরকার। পরিগণকে সংযোজনী ভাষা থেকে যান্ত্রিক ভাষাতে অনুবাদের কাজ করে যে গঠনরূপায়ক (System software) সফট্ওয়্যার তাকে বলা হয় সংযোজক—নিয়ন্ত্রক (Assembler)। এর কাজ হল সংযোজনী ভাষায় লেখা ক্রিয়াদেশলিপিকে যান্ত্রিক ভাষায় অনুবাদ করা।

উচ্চমানের ভাষা (High Level Language) ঃ

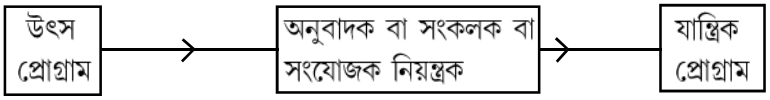
যান্ত্রিক ভাষা বা সংযোজনী ভাষায় কোন সুদীর্ঘ ক্রিয়াদেশলিপি তৈরী করা যথেষ্ট সময় সাপেক্ষ এবং শ্রমসাপেক্ষ কাজ। এই কাজে যথেষ্ট দক্ষতা এবং পরিগণকের আভ্যন্তরীণ বিন্যাসজ্ঞান থাকা আবশ্যিক। এই সমস্ত কারণে গোড়ার দিকে পরিগণকের ব্যবহার শুধুমাত্র বাস্তুকার, যন্ত্র শিল্পী ও বিজ্ঞানী মহলে সীমাবদ্ধ ছিল। ক্রমশঃ বৈদ্যুতিন্ প্রযুক্তিবিদ্যার অগ্রগতির সাথে সাথে পরিগণকীয় কর্মভাষাকে সাধারণ মানুষের ব্যবহারোপযোগী করে তুলতে নিরন্তর প্রয়াস চালিয়ে গেছেন বিজ্ঞানীরা এবং পরিগণকীয় ভাষাবিদেরা সম্মিলিতভাবে। এর ফলস্বরূপ, বিগত কয়েক দশকে বেশ কিছু উন্নতমানের পরিগণকীয় কর্মভাষার উদ্ভাবন হয়েছে। উদাহরণস্বরূপ, BASIC, FORTRAN, C, PASCAL, ALGOL, PL/1 ইত্যাদি ভাষার নামোল্লেখ করা যেতে পারে। এইসব উচ্চমানের ভাষাতে চলতি ইংরেজী ভাষার বিভিন্ন শব্দের ব্যবহার রয়েছে। এর ফলে এই সমস্ত উচ্চমানের ভাষা সাধারণ মানুষের পক্ষে, অনেক বেশী গ্রহণযোগ্য হয়েছে। বর্তমানে শুধুমাত্র গাণিতিক হিসেব সংক্রান্ত কাজের জন্য পরিগণকের ব্যবহার সীমিত নেই। বিভিন্ন ক্ষেত্রে প্রয়োজনভিত্তিক বিভিন্ন উচ্চমানের পরিগণকীয় ভাষা উদ্ভাবিত হয়েছে। এইসব উচ্চমানের ভাষা ক্রিয়াদেশ লিপিকরণের কাজটিকে সাধারণ মানুষের পক্ষেও সহজ করে দিয়েছে।

পরিগণক যেহেতু শুধুমাত্র যান্ত্রিক ভাষায় লেখা নির্দেশ পালনে সক্ষম সেহেতু উচ্চমানের ভাষায় লেখা কোন

ক্রিয়াদেশলিপি প্রথমে যান্ত্রিক ভাষায় অনুবাদিত হয়। উচ্চমানের ভাষা থেকে যান্ত্রিক ভাষায় রূপান্তর বা অনুবাদের কাজটি করে কোন সংকলক (Compiler) বা কোন অনুবাদক (Interpreter)। এই সংকলক বা অনুবাদক হল আসলে এক একটি গঠন রূপায়ক সফট্ওয়্যার (Systerm Software)।

উৎস প্রোগ্রাম এবং যান্ত্রিক প্রোগ্রাম (Source Program and Object Program) ঃ

উচ্চমানের ভাষায় অথবা সংযোজনী ভাষায় লেখা কোন প্রোগ্রাম বা ক্রিয়াদেশলিপিকে বলা হয় উৎস প্রোগ্রাম। কোন উৎস প্রোগ্রামকে কার্যকর করে তোলার জন্য সেটিকে প্রথমে কোন উপযুক্ত সংকলন বা অনুবাদক বা সংযোজক নিয়ন্ত্রকের সাহায্যে যান্ত্রিক ভাষায় রূপান্তর করে নিতে হয়। যান্ত্রিক ভাষায় লেখা উৎস প্রোগ্রামের সমতুল্য প্রোগ্রামটিকে বলা হয় যান্ত্রিক প্রোগ্রাম।



সংকলক এবং অনুবাদক (Complier and Interpreter) ঃ

সংকলক এর কাজ হল উচ্চমানের ভাষায় লেখা প্রোগ্রামকে সমতুল্য যান্ত্রিক প্রোগ্রামে রূপান্তর করা। যেমন, FORTRAN ভাষায় লেখা কোন প্রোগ্রামকে কোন পরিগণককে কার্যকর করতে হলে বা চালাতে হলে ঐ পরিগণকটির আভ্যন্তরীণ বিন্যাসে অবশ্যই FORTRAN—সংকলককে অন্তর্ভূক্ত থাকতে হবে বা করে নিতে হবে। সাধারণতঃ পরিগণক উৎপাদক সংস্থা থেকে প্রয়োজনীয় সংকলক বা অনুবাদক সরবরাহ করা হয়। সংকলক এর কার্যবিধি কয়েকটি ধাপে সম্পাদিত হয়। যেমন—

- সংকলক এর প্রথম কাজ হল উৎস প্রোগ্রামটি আগাগোড়া পরীক্ষা করা এবং প্রোগ্রামটিতে ব্যবহৃত নির্দেশাদিতে পদবিন্যাস ব্যাকরণসম্মত না থাকলে সেগুলি চিহ্নিত করা।
- চিহ্নিত ত্রুটিগুলি সম্পর্কে প্রোগ্রাম রচয়িতা যাতে ওয়াকিবহাল হতে পারেন এবং সংশোধন করতে পারেন সেজন্য ভাষা সংকলক প্রোগ্রামে বর্তমান ত্রুটিগুলির জন্য একটি ত্রুটির প্রকৃতি নিরূপক প্রোগ্রাম তালিকা তৈরী করে। ত্রুটির প্রকৃতি নিরূপক প্রোগ্রামগুলি প্রতিটি উচ্চমানের ভাষায় লেখা, যাতে প্রোগ্রাম রচয়িতা সহজেই তার ত্রুটিগুলি অনুধাবন করতে পারেন।
- যদি উৎস প্রোগ্রামে ভাষার ব্যাকরণজনিত ত্রুটি না থাকে তাহলে ভাষাসংকলকের কাজ হল উৎস প্রোগ্রামটিকে সমতুল্য যান্ত্রিক প্রোগ্রামে রূপান্তরিত করা।

অনুবাদক এর কার্যপ্রণালী কিছুটা সংকলক এর মত। তফাৎ এই যে, সংকলক প্রথমে উৎসপ্রোগ্রামটি আগাগোড়া পড়ে নেয় এবং তারপর ব্যাকরণজনিত ত্রুটিতালিকা তৈরী করে অথবা ত্রুটি না পেলে উৎস-প্রোগ্রামের সমতুল্য

যান্ত্রিক প্রোগ্রাম তৈরী করে। কিন্তু অনুবাদকের কাজ হল উৎস-প্রোগ্রামের নির্দেশগুলি একটি একটি করে পড়া এবং তাতে ব্যাকরণজনিত ত্রুটি আছে কিনা তা পরীক্ষা করে দেখা; যদি কোন ত্রুটি না থাকে তাহলে নির্দেশটি তৎক্ষণাৎ যান্ত্রিক ভাষাতে রূপান্তরিত করা এবং নির্দেশটি কার্যকর করা। একটি নির্দেশ কার্যকর করবার পরই অনুবাদক পরবর্তী নির্দেশবাক্য পাঠ করতে পারে।

সংকলক এবং অনুবাদক এর কার্যপ্রণালী থেকে বুঝতে অসুবিধা হয় না যে, সংকলক এর সাহায্যে কোন সুদীর্ঘ প্রোগ্রাম চালাতে অপেক্ষাকৃত কম সময় লাগবে। কারণ, সংকলকে এক একটি নির্দেশবাক্যের জন্য একবারই ভাষা রূপান্তর প্রক্রিয়ার প্রয়োজন হয়। কিন্তু অনুবাদকের ক্ষেত্রে কোন একটি নির্দেশকে একাধিকবার কার্যকর করতে হলে প্রতিবারেই ভাষা রূপান্তর প্রক্রিয়ার প্রয়োজন হবে। তবে অনুবাদকের দুটি সুবিধাজনক দিকও আছে। প্রথমতঃ অনুবাদক এর জন্য প্রয়োজনীয় বৈদ্যুতিন্ প্রযুক্তিবিদ্যা অনেক সরল। দ্বিতীয়তঃ অনুবাদক চিহ্নিত ত্রুটি নির্দেশগুলি সংশোধন করা অপেক্ষাকৃত সহজ।

কয়েকটি উল্লেখযোগ্য উচ্চমানের ভাষা ঃ

বিগত পাঁচ দশকে বেশ কিছু উচ্চমানের ভাষার উদ্ভাবন হয়েছে। বিভিন্ন প্রয়োগক্ষেত্রের বিভিন্ন প্রয়োজনীয়তার উপর লক্ষ্য রেখে বিভিন্ন উচ্চমানের ভাষার সৃষ্টি হয়েছে। নীচে কয়েকটির উল্লেখ করা হল।

BASIC ঃ BASIC শব্দটির পূর্ণরূপ হল Begineers' All-purpose Symbolic Instruction Code এই ভাষাটির স্রষ্ঠা হলেন জন কেমেনি (John Kemeny) এবং ই. কুর্টজ্ (E. Kurtz) যৌথভাবে। ভাষাটির সৃষ্টি হয়েছিল 1963-64 সালে। এই ভাষার ব্যাকরণ খুবই সহজ ও সরল। গাণিতিক সমস্যা সমাধানের জন্য যে সমস্ত নির্দেশ BASIC ভাষাতে ব্যবহার করা হয় সেগুলি আমাদের সাধারণ ব্যবহারিক ক্ষেত্রের সাথে যথেষ্ট সাদৃশ্যপূর্ণ। ম্যাট্রিক্স সংক্রান্ত বীজগণিত, অসাংখ্যিক তথ্যকোষাঙ্ক নিয়ে কিছু উদ্দেশ্যমূলক কাজ, চিত্রাঙ্কণ ইত্যাদি বিভিন্ন ধরণের কাজের জন্য BASIC ভাষাতে প্রোগ্রাম তৈরী করা যায়।

FORTRAN ঃ FORTRAN শব্দটির অর্থ হল FORmula TRANslation অর্থাৎ সূত্র রূপান্তর বা সূত্র ভাষান্তর। এই ভাষাটির উদ্ভাবন হয়েছিল 1956 সালে International Business Machines Corporation (সংক্ষেপে IBM) সংস্থার তত্ত্বাবধানে। পূর্তবিদ্যা এবং বিজ্ঞান সংক্রান্ত বিভিন্ন সমস্যা সমাধানের জন্য প্রয়োজনীয় প্রোগ্রাম লেখার পক্ষে এই ভাষাটি বিশেষভাবে উপযোগী। যে কোন প্রকার গাণিতিক সমস্যা সমাধানের জন্য FORTRAN ভাষাতে প্রোগ্রাম লেখা সম্ভব। কিন্তু অসাংখ্যিক তথ্যকোষাঙ্ক, যেমন—ব্যক্তির নাম, ঠিকানা ইত্যাদি তথ্যনির্ভর কাজ করার জন্য FORTRAN ভাষা খুব একটা সুবিধাজনক নয়। গোড়ার দিকে FORTRAN ভাষা কিছুটা যন্ত্রনির্ভর ছিল। পরে বেশ কয়েকটি ধাপে এই ভাষা ক্রমশঃ উন্নীত হয়েছে। 1960 সালে প্রথম Americal National Standards Institution সংক্ষেপে ANSI সংস্থা এই ভাষাটির এক প্রামাণ্য রূপ প্রকাশ করে। FORTRAN 4, FORTRAN 77, FORTRAN 90, FORTRAN 95 ইত্যাদি হল এই ভাষাটির বিভিন্ন পর্যায়ের ক্রমউন্নীত রূপ। এদের মধ্যে FORTRAN 77 এর জনপ্রিয়তা সর্বাধিক। ভাষাটির উন্নীতকরণের সময় প্রতিটি পর্যায়ে তার পূর্ববর্তী পর্যায়ে ব্যবহৃত সমস্ত নির্দেশ একই রখা হয়েছে।

COBOL ঃ Common Business Oriented Language এর সংক্ষিপ্ত রূপ হল COBOL। এই ভাষাটির

জন্ম 1959 সালে United States Department of Defence এর তত্ত্বাবধানে। ভাষাটির নাম থেকে স্পষ্টতঃ প্রতীয়মান যে, বাণিজ্যিক হিসেবসংক্রান্ত কাজের জন্য উপযুক্ত ভাষা COBOL। সাংখ্যিক এবং অসাংখ্যিক উভয়প্রকার তথ্য নিয়ে কাজ করা যেতে পারে COBOL ভাষাতে। এই ভাষাটি FORTRAN ভাষার মত গাণিতিক হিসেবকার্যে সুদক্ষ নয়। বরং বলা যায়, অসাংখ্যিক তথ্যনির্ভর কাজে FORTRAN ভাষার যে দুর্বলতা আছে সেই ঘাটতি পূরণ করতে পারে COBOL।

সংক্ষিপ্তসার ঃ

মানুষ যেমন মনের ভাব প্রকাশ করার জন্য ভাষা ব্যবহার করে ঠিক তেমনি পরিগণক যন্ত্রও নির্দিষ্ট কিছু ভাষা ব্যবহার করে। পরিগণক যন্ত্রে প্রধানত দুই ধরণের ভাষা—নিম্নমানের ভাষা ও উচ্চমানের ভাষা ব্যবহৃত হয়। এই পাঠে নিম্নমানের ভাষা (যান্ত্রিক ভাষা ও সংযোজনী ভাষা) ও উচ্চমানের ভাষা সম্বন্ধে সংক্ষেপে আলোচনা করা হয়েছে। উচ্চমানের ভাষাকে কিভাবে অনুবাদক বা সংকলক বা সংযোজকের সাহায্যে যান্ত্রিক ভাষায় রূপান্তর করা হয় তার বর্ণনাও এখানে করা হয়েছে। তিনটি উচ্চমানের ভাষা—BASIC, FORTRAN ও COBOL সম্বন্ধে সংক্ষেপে আলোচনা করা হয়েছে।

একক 4 ❑ ক্রিয়ানুক্রম তালিকা (Algorithm) ও কর্মপ্রবাহ চিত্র (Flow Chart)

এককের উদ্দেশ্য ঃ

এই একক অনুধাবন করলে পাঠক নিম্নলিখিত বিষয়গুলি সম্বন্ধে জ্ঞাত হবে।

- ক্রিয়ানুক্রম তালিকা
- ক্রিয়ানুক্রম তালিকার বৈশিষ্ট্য
- কর্মপ্রবাহ চিত্র
- কর্মপ্রবাহ চিত্রে ব্যবহৃত চিহ্ন
- কর্মপ্রবাহ চিত্রের ব্যবহার

ক্রিয়ানুক্রম তালিকা (Algorithm) ও কর্মপ্রবাহ চিত্র (Flow Chart)

পরিগণকের সাহায্যে গাণিতিক বা অ-গাণিতিক বিভিন্ন ধরণের সমস্যা সুষ্ঠুভাবে সম্পাদন করা যায়। কোন সুদীর্ঘ পুনরাবৃত্তি মূলক কাজ পরিগণকের মাধ্যমে অনায়াসে করা সম্ভবপর। কিন্তু পরিগণকের নিজস্ব এমন কোন ক্ষমতা নেই যা দিয়ে সে কোন সমস্যা সমাধানের জন্য আবশ্যিক অনুসরণীয় ধাপগুলি নিজে থেকে স্থির করতে পারে। কাজেই পরিগণকের সাহায্য নেওয়ার আগে প্রাথমিক এই কাজটি অর্থাৎ ক্রিয়ানুক্রম তালিকা তৈরীর কাজটি করতে হয় ক্রিয়াদেশলিপিকারককে। সুতরাং ক্রিয়াদেশলিপি তৈরীর পূর্বের ধাপ হল ক্রিয়ানুক্রম তালিকা তৈরী করা।

ক্রিয়ানুক্রম তালিকা (Algarithm) ঃ কোন একটি বিশেষ সমস্যা সমাধানের জন্য ক্রিয়ানুক্রম তালিকা তৈরীর জন্য লিপিকারককে নিম্নলিখিত কয়েকটি বিষয়ে মনোযোগ দিতে হয়—

- সমস্যাটি সুনির্দিষ্টভাবে বুঝে নিতে হবে।
- সমস্যাটি কিভাবে সমাধান করা যেতে পারে তা বিশ্লেষণ করে দেখতে হবে। যদি একাধিক উপায়ে সমাধান সম্ভবপর হয় তাহলে যে উপায়টি পরিগণকের পক্ষে সর্বাধিক কার্যকরী হবে সেই উপায়টি অবলম্বন করতে হবে।
- সমস্যা সমাধানের জন্য প্রয়োজনীয় ধাপগুলি সুচিহ্নিত করতে হবে এবং তাদের মধ্যে যে ধারাবাহিকতা থাকা আবশ্যিক সেই ক্রিয়ানুক্রম অনুধাবন করতে হবে।
- সুনির্দিষ্ট ক্রিয়ানুক্রমের অন্তবর্তী কোন পর্যায়ে একাধিক সম্ভাবনা থাকলে প্রতিটি সম্ভাবনাকে বিচার্য বিষয় হিসাবে দেখতে হবে।

উপরিল্লিখিত বিষয়গুলি মেনে ক্রিয়াদেশলিপিকারক প্রথমে একটি ক্রিয়ানুক্রম তালিকা বা এ্যালগোরিদম্

(Algorithm) তৈরী করেন। এ্যালগোরিদম্ শব্দটি এসেছে নবম শতাব্দীর বিখ্যাত আরব গণিতজ্ঞ Abu Jafar Mohammed Bin Musa Al-khowarizmi-এর নাম থেকে। এ্যালগোরিদম্ লেখা যেতে পারে ইংরাজী বা অন্য কোন কথপোকথনের ভাষাতে অথবা কোন পরিগণকীয় ভাষাতে। তবে বলা বাহুল্য যে, প্রথমেই পরিগণকীয় ভাষাতে এ্যালগোরিদম্ লিখতে যাওয়া সম্পূর্ণ অর্থহীন। কারণ পরিগণকীয় ভাষায় প্রোগ্রাম লেখার কাজকে সহজ করবার উদ্দেশ্যেই ক্রিয়ানুক্রম তালিকা তৈরী জরুরী। বলা যায়, পরিগণকীয় প্রোগ্রাম তৈরীর প্রাক্ প্রস্তুতি হল এ্যালগোরিদম তৈরী।

যে কোন ক্রিয়ানুক্রম তালিকাতে আমরা নিম্নলিখিত বৈশিষ্ট্যগুলি দেখতে পাই—

(i) বাইরে থেকে তথ্য সরবরাহ (Input) ঃ— প্রতিটি এ্যালগোরিদমের শুরুতে কিছু তথ্য পরিগণককে সরবরাহ করতে হয়।

(ii) প্রক্রিয়া শেষে উদ্দিষ্ট তথ্য প্রাপ্তি (Output) ঃ— প্রতিটি এ্যালগোরিদমের শেষে এক বা একাধিক তথ্য পরিগণক থেকে পাওয়া যায়।

(iii) দ্ব্যর্থহীন ক্রিয়ানির্দেশ ঃ— এ্যালগোরিদমে ব্যবহৃত ক্রিয়ানির্দেশগুলি কখনওই পরস্পর বিরোধী হতে পারে না এবং প্রতিটি ক্রিয়ানির্দেশ হবে দ্ব্যর্থহীন।

(iv) সীমিত সংখ্যক ক্রিয়ানির্দেশ ঃ— প্রতিটি এ্যালগোরিদমে সীমিত সংখ্যক ক্রিয়ানির্দেশ থাকতে পারে যাতে সম্পূর্ণ প্রক্রিয়াটি সীমিত সময়ে সম্পাদিত হতে পারে।

(v) ক্রিয়ানির্দেশাবলীর সম্পূর্ণতা ঃ— এ্যালগোরিদমে উল্লিখিত ক্রিয়ানির্দেশাবলীর স্বয়ং সম্পূর্ণ হওয়া বাঞ্ছনীয় যাতে সেটি কোন বিশেষ ধরণের যে কোন সমস্যা সমাধানের জন্য উপযুক্ত হতে পারে।

নীচে কয়েকটি সমস্যা এবং সমাধানের জন্য প্রয়োজনীয় ক্রিয়ানুক্রম তালিকা দেওয়া হল।

উদাহরণ-1 ঃ

সমস্যা ঃ x এর প্রদত্ত মানের জন্য $\sin x$ এর মান নির্ণয় কর।

এ্যালগোরিদম ঃ

1 ম ধাপ ঃ x এর মান পড়।

2 য় ধাপ ঃ $y = \sin x$ এর মান নির্ণয় কর।

3 য় ধাপ ঃ x ও y এর মান লেখ। এবার প্রক্রিয়াটি সমাপ্ত কর।

উদাহরণ-2 ঃ

সমস্যা ঃ প্রদত্ত তিনটি ভিন্ন সংখ্যা x, y ও z এর মধ্যে সবচেয়ে বড় সংখ্যাটি নির্ণয় কর।

এ্যালগোরিদম্ ঃ

1ম ধাপ ঃ x, y এবং z এর মান পড়।

2য় ধাপ ঃ যদি $x > y$ হয় তাহলে 3য় ধাপ অনুসরণ কর। অন্যথায় 4র্থ ধাপ অনুসরণ কর।

3য় ধাপ ঃ যদি $x > z$ হয়, তাহলে সবচেয়ে বড় সংখ্যাটি হল x। x এর মান লেখ এবং প্রক্রিয়াটি সমাপ্ত কর। নতুবা সবচেয়ে বড় সংখ্যাটি হল z। z এর মান লেখ এবং প্রক্রিয়াটি সমাপ্ত কর।

4র্থ ধাপ ঃ যদি $z > y$ হয়, তাহলে সবচেয়ে বড় সংখ্যাটি হল z। z এর মান লেখ এবং প্রক্রিয়াটি সমাপ্ত কর। না হলে সবচেয়ে বড় সংখ্যাটি হল y। y এর মান লেখ এবং প্রক্রিয়াটি সম্পন্ন কর।

মন্তব্য ঃ এই এ্যালগোরিদমের যুক্তি ব্যবহার করে প্রদত্ত কয়েকটি সংখ্যার মধ্যে সবচেয়ে বড় সংখ্যাটি খুঁজে বের করা যেতে পারে অথবা সংখ্যাগুলিকে মানের অধঃক্রমানুসারে বা ঊর্ধ্বক্রমানুসারে সাজানো যেতে পারে।

উদাহরণ-3 ঃ

সমস্যা ঃ প্রদত্ত দুটি ভিন্ন ধনাত্মক পূর্ণসংখ্যা p ও q এর গ.সা.গু নির্ণয় কর।

এ্যালগোরিদম্ ঃ

1ম ধাপ ঃ p ও q এর মান পড়।

2য় ধাপ ঃ যদি $p < q$ হয় তাহলে p ও q এর মান পরস্পরের মধ্যে বদল করে নাও।

3য় ধাপ ঃ q দ্বারা p কে ভাগ কর এবং ভাগশেষ r কে নির্ণয় কর।

(r হল একটি পূর্ণসংখ্যা এবং $0 \le r \le q-1$)

4র্থ ধাপ ঃ যদি $r = 0$ হয়, তাহলে নির্ণেয় গ.সা.গু হল q। q এর মান লেখ এবং প্রক্রিয়াটি সমাপ্ত কর। যদি $r = 1$ হয়, তাহলে সংখ্যা দুটি পরস্পর মৌলিক বলে লেখ এবং প্রক্রিয়াটি সমাপ্ত কর। যদি $r > 1$ হয়, তাহলে পরবর্তী ধাপে যাও।

5ম ধাপ ঃ p ও q এর মান যথাক্রমে q ও r এর মান দ্বারা প্রতিস্থাপিত কর এবং 3য় ধাপে যাও। যতক্ষণ $r > 1$ থাকছে ততক্ষণ প্রক্রিয়াটির পুনরাবৃত্তি কর।

উদাহরণ-4 ঃ

সমস্যা ঃ $ax^2 + bx + c = 0$ এই দ্বিঘাত সমীকরণটি সমাধান কর, a, b ও c এর বাস্তব মানের জন্য।

এ্যালগোরিদম্ ঃ

1ম ধাপ ঃ a, b ও c এর মান পড়।

2য় ধাপ ঃ যদি $a = 0$ এবং $b = 0$ হয়, তাহলে লেখ 'No root exists' এবং প্রক্রিয়াটি সমাপ্ত কর। অন্যথায় পরবর্তী ধাপ অনুসরণ কর।

3য় ধাপ ঃ যদি $a = 0$ এবং $b \neq 0$ হয়, তাহলে লেখ 'Equation is linear'। সমীকরণের একমাত্র বীজটি নির্ণয় কর ঃ Root $= -\frac{c}{b}$। নির্ণেয় বীজটির মান লেখ এবং প্রক্রিয়াটি সমাপ্ত কর। অন্যথায় পরবর্তী ধাপ অনুসরণ কর।

4র্থ ধাপ ঃ সমীকরণের নিরূপকটি নির্ণয় কর ঃ $d = b^2 - 4ac$

5ম ধাপ ঃ যদি $d = 0$ হয়, তাহলে লেখ 'Roots are equal'। সমীকরণের সমমানের বীজ দুটি নির্ণয় কর ঃ

Root 1 $= -\frac{b}{2a}$, Root 2 $= -\frac{b}{2a}$ । Root 1 এবং Root 2 এর মান লেখ এবং প্রক্রিয়াটি সমাপ্ত কর। অন্যথায় পরবর্তী ধাপ অনুসরণ কর।

6ষ্ঠ ধাপ ঃ যদি $d > 0$ হয়, তাহলে লেখ 'Roots are real and unequal'। সমীকরণের বীজদ্বয় নির্ণয় কর ঃ

Root 1 $= \frac{-b+\sqrt{d}}{2a}$, Root 2 $= \frac{-b-\sqrt{d}}{2a}$ । Root 1 ও Root 2 এর মান লেখ এবং প্রক্রিয়াটি সমাপ্ত কর। অন্যথায় পরবর্তী ধাপে যাও।

7ম ধাপ ঃ যদি $d < 0$ হয়, তাহলে লেখ 'Roots are imaginary'। জটিল বীজদ্বয়ের বাস্তব ও কাল্পনিক অংশ নির্ণয় কর ঃ Realpart $= -\frac{b}{2a}$, Imagpt $= \frac{\sqrt{-d}}{2a}$ । Realpt ও Imagpt এর মান লেখ এবং প্রক্রিয়াটি সমাপ্ত কর।

উদাহরণ-5 ঃ

সমস্যা ঃ বিদ্যালয়ের নবম শ্রেণীর ছাত্রদের গড় উচ্চতা নির্ণয় কর।

এ্যালগোরিদম্ ঃ

1ম ধাপ ঃ নবম শ্রেণীর মোট ছাত্রসংখ্যাকে N প্রচল (parameter) দ্বারা সূচিত কর। N এর মান পড়ে নাও।

2য় ধাপ ঃ মোট উচ্চতার প্রারম্ভিক মান ধর শূন্য ঃ Totalh = 0 এবং ছাত্রসংখ্যা গণনা শুরু করতে ধর ঃ count = 1

3য় ধাপ ঃ ছাত্রের উচ্চতা পড় (জেনে নাও)। ধর, উচ্চতা হল h, যা একটি চলরাশি। h এর মান পড়।

4র্থ ধাপ ঃ মোট উচ্চতার আংশিক যোগফল নির্ণয় কর ঃ Total h = Total h + h

(মোট উচ্চতা = পূর্ববর্তী মোট উচ্চতা + আলোচ্য ছাত্রটির উচ্চতা

অর্থাৎ, যোগফলের নতুন মান = যোগফলের বর্তমান মান + চলরাশির বর্তমান মান)

5ম ধাপ ঃ যদি Count = N হয়, তাহলে 6ষ্ঠ ধাপে যাও। নাহলে, count এর মান পরিবর্তন কর ঃ Count = Count + 1। এরপর 3য় ধাপে যাও।

6ষ্ঠ ধাপ ঃ গড় উচ্চতা নির্ণয় কর ঃ Average h = Total h/N। Average h এর মান লেখ এবং প্রক্রিয়াটি সমাপ্ত কর।

মন্তব্য ঃ (i) এই এ্যালগোরিদম্টিতে ছাত্রদের উচ্চতা এবং N এর মান হল পরিগণকে সরবরাহিত তথ্য (বা input) এবং ছাত্রদের গড় উচ্চতা হল প্রক্রিয়াশেষে প্রাপ্ত তথ্য (বা output)।

(ii) এই এ্যালগোরিদম্টির 3য়, 4র্থ এবং 5ম ধাপকে পুনরাবৃত্তি করতে হচ্ছে। এই ধরনের পুনরাবৃত্তিমূলক নির্দেশাদি তৈরী করে একটি লুপ্ (loop)।

(iii) এ্যালগোরিদমে কোন লুপ্ বর্তমান থাকলে লুপ্টির পুনরাবৃত্তি কতবার হবে তার সুস্পষ্ট নির্দেশও এ্যালগোরিদম্টিতে থাকতে হবে। নতুবা গোটা প্রক্রিয়াটি কখনো সমাপ্ত হতে পারবে না।

(iv) কোন একটি এ্যালগোরিদমে একাধিক লুপ্ থাকতে পারে। যদি একাধিক লুপ্ থাকে তবে এই লুপ্গুলি থাকতে পারে দু'ভাবে—

(ক) লুপ্গুলি পরস্পর বিচ্ছিন্নভাবে থাকতে পারে। অর্থাৎ, বিভিন্ন লুপের অন্তর্নিহিত প্রক্রিয়াগুলি স্বাধীনভাবে সম্পাদিত হতে পারে।

(খ) একটি লুপের ক্রিয়াক্ষেত্রের মধ্যে আরও একটি লুপ্ ক্রিয়াশীল থাকতে পারে, তবে দুটি লুপ্ের ক্রিয়াক্ষেত্র কখনই পরস্পরছেদী হবে না।

এখানে লক্ষ্যনীয় একটি বিষয় হলো 4র্থ বা 5ম ধাপে এমন একটি নির্দেশ ব্যবহার করা হয়েছে যার সাধারন রূপটি হলো $x = x + k$ যেমন, $x = x + 3$ । প্রথাগত গণিতশাস্ত্রে এর কোন অর্থ হয় না। তবে এ্যালগোরিদমের ভাষায় বা পরিগণকীয় ভাষাতে এর অর্থ হল x–চল্টির বর্তমান মানের সঙ্গে k যোগ করে প্রাপ্ত যোগফলটির মান্ দিয়ে x-এর বর্তমান মানটিকে প্রতিস্থাপন করা।

কর্মপ্রবাহ চিত্র/ক্রিয়াপ্রবাহ চিত্র/নির্দেশ প্রবাহ চিত্র (Flow Chart) ঃ

ক্রিয়াপ্রবাহ চিত্র হল ক্রিয়ানুক্রম তালিকার সচিত্র উপস্থাপনা। পরিগণকীয় ভাষাতে কোন প্রোগ্রাম লেখার আগে প্রোগ্রাম রচয়িতা প্রথমে সমস্যা সমাধানের জন্য প্রয়োজনীয় নির্দেশাদিকে একটি এ্যালগোরিদম্-এর মাধ্যমে লিপিবদ্ধ করতে পারেন। এতে প্রোগ্রাম লেখার কাজ অনেকটা সহজ হয়। কিন্তু সমস্যা কিছুটা জটিল হলে অর্থাৎ এ্যালগোরিদমে একাধিক লুপ্ বর্তমান থাকলে বা অন্তবর্তী বিভিন্ন ধাপে একাধিক যৌক্তিক সম্ভাবনা (logical possibility) থাকলে এ্যালগোরিদম্ আর ততটা সহজবোধ্য থাকে না। এক্ষেত্রে এ্যালগোরিদমের সচিত্র উপস্থাপনা অর্থাৎ ক্রিয়াপ্রবাহ চিত্র বিষয়টিকে সহজভাবে বুঝতে সাহায্য করে।

একটি ক্রিয়াপ্রবাহ চিত্রে প্রয়োজনীয় ক্রিয়ানির্দেশাদি বিভিন্ন ধরনের জ্যামিতিক আকারের মধ্যে লেখা হয়। এই জ্যামিতিক আকারগুলি পরস্পর দিক নির্দেশক রেখাংশ দ্বারা সংযুক্ত থাকে। ক্রিয়া প্রবাহচিত্রে ব্যবহৃত বিভিন্ন জ্যামিতিক আকার বা নকশাগুলি ANSI কর্তৃক প্রামাণ্যরূপে স্বীকৃত এবং গৃহীত হয়েছে। যে সমস্ত জ্যামিতিক নকশার বহুল ব্যবহার আমরা ক্রিয়া প্রবাহচিত্রে দেখতে পাই সেগুলির বর্ণনা দেওয়া হল নীচে—

(1) প্রান্তিক চিহ্ন (Terminal Symbols) ঃ

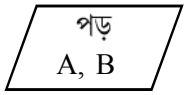
ক্রিয়া প্রবাহচিত্রে শুরু এবং শেষ—এই দুই প্রান্ত বোঝানে হয় উপবৃত্তাকার ক্ষেত্র দ্বারা। যেমন—ক্রিয়াপ্রবাহের শুরু বোঝান হয়।

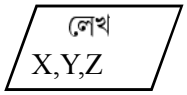
চিহ্ন দ্বারা এবং ক্রিয়া প্রবাহের সমাপ্তি বোঝান হয়।

চিহ্ন দ্বারা। এছাড়া কোন একটি ধাপ থেকে বিশেষ কোন ধাপে ফিরে যাওয়ার দরকার হলে আঁকা হয়।

(2) তথ্য আদান-প্রদান ক্ষেত্রচিহ্ন (Input/Output Box) ঃ

ক্রিয়া প্রবাহচিত্রে কোন ধাপে তথ্য সরবরাহ করা হচ্ছে বোঝাতে বা প্রক্রিয়াশেষে কোন তথ্য প্রাপ্তি হচ্ছে বোঝাতে সামান্তরিক চিহ্ন ব্যবহার করা হয়। যেমন—





(3) প্রক্রিয়াকরণ ক্ষেত্রচিহ্ন ঃ

(Processing Box)

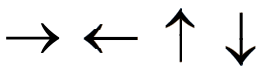
ক্রিয়া প্রবাহচিত্রে কোন ধাপে তথ্যনির্ভর উদ্দেশ্যমূলক কাজ (যেমন, গাণিতিক হিসাব, সাংখ্যিক বা অসাংখ্যিক তথ্য প্রক্রিয়াকরণ ইত্যাদি) সাধিত হচ্ছে তা বোঝানোর জন্য আয়তক্ষেত্র ব্যবহার করা হয়। যেমন—

$X = 2 \cdot 5$	অথবা	$X \leftarrow 2 \cdot 5$
$S = S + A$	অথবা	$S \leftarrow S + A$
$F = A + 2B - C^2$	অথবা	$F \leftarrow A + 2B - C^2$

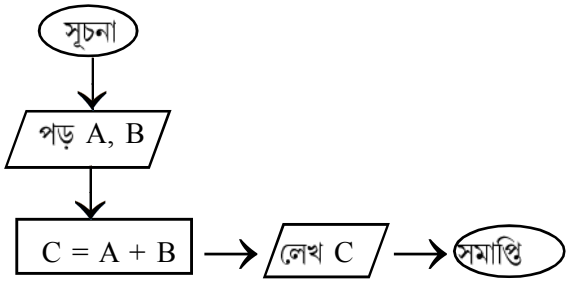
এখানে $X = 2 \cdot 5$ বা $X \leftarrow 2 \cdot 5$ এর অর্থ হল 2·5 সংখ্যাটি X চলরাশির মান হিসাবে জমা কর। $S = S + A$ বা $S \leftarrow S + A$ এর অর্থ হল $(S + A)$ এই যোগফলটি দ্বারা S এর বর্তমান মানকে অপসারিত কর।

(4) প্রবাহ দিক নির্দেশক রেখা ঃ

(Flow lines)

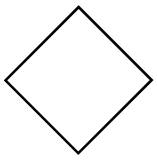


ক্রিয়া প্রবাহচিত্রে দিক নির্দেশক রেখাংশ বা তীরচিহ্ন দ্বারা ক্রিয়াপ্রবাহের অভিমুখ সূচিত করা হয়। যেমন—

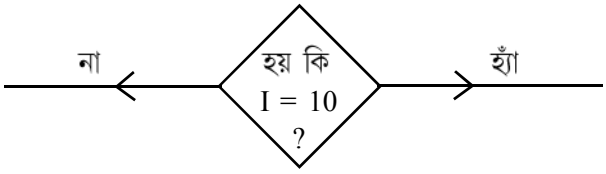


(5) সিদ্ধান্ত নির্ণয়ক ক্ষেত্রচিহ্ন ঃ

(Decision Box)



ক্রিয়া প্রবাহচিত্রে কোন ধাপে তুলনামূলক বিচার বা সিদ্ধান্ত নির্ণয়ক পরিস্থিতিকে বোঝান হয় রম্বস এর মধ্যে। যেমন—

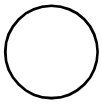


(6) সম্পর্ক স্থাপক চিহ্ন/সম্বন্ধ স্থাপক চিহ্ন ঃ

(Connector)

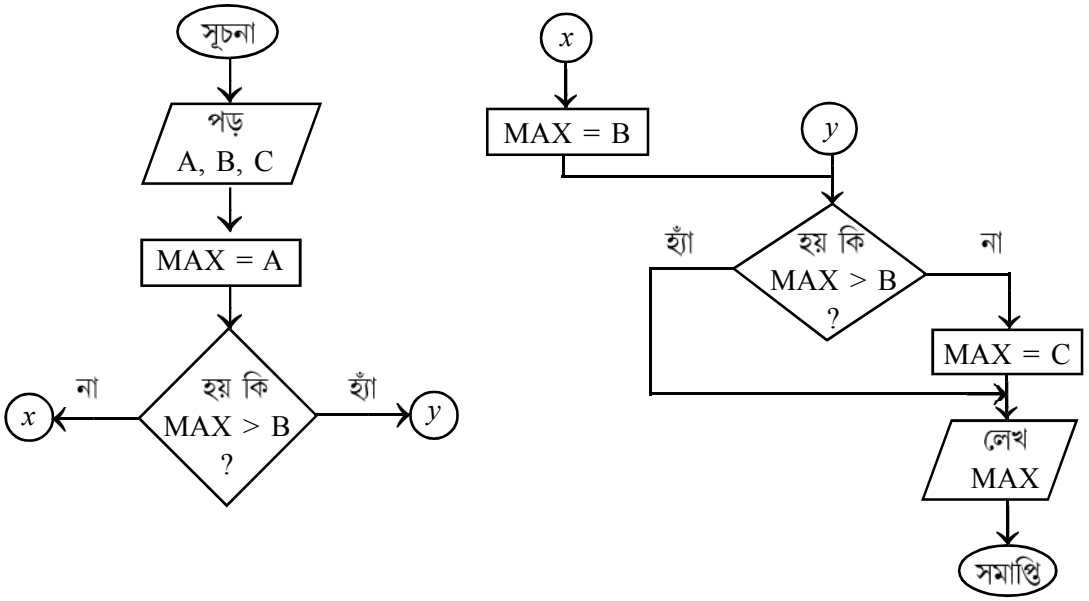
সম্পর্ক স্থাপক চিহ্ন দু'ধরণের।

(ক) একই পৃষ্ঠায় সম্পর্ক স্থাপক চিহ্ন ঃ

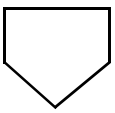


(On-page Connector)

ক্রিয়াপ্রবাহ চিত্রের অংশবিশেষকে একই পৃষ্ঠার অন্য কোন স্থানে এঁকে দেখানো যেতে পারে। এজন্য আংশিক ক্রিয়াপ্রবাহচিত্রের শুরুতে একটি চিহ্নিত বৃত্ত আঁকতে হবে এবং অনুরূপ চিহ্নযুক্ত আর একটি বৃত্ত আঁকতে হবে ঠিক যেখানে আংশিক ক্রিয়া প্রবাহচিত্রটি সম্পর্কযুক্ত হবে মূল ক্রিয়াপ্রবাহচিত্রের সাথে। যেমন—



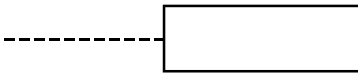
(খ) ভিন্ন পৃষ্ঠায় সম্পর্ক স্থাপক চিহ্ন ঃ



(Off-page Connector)

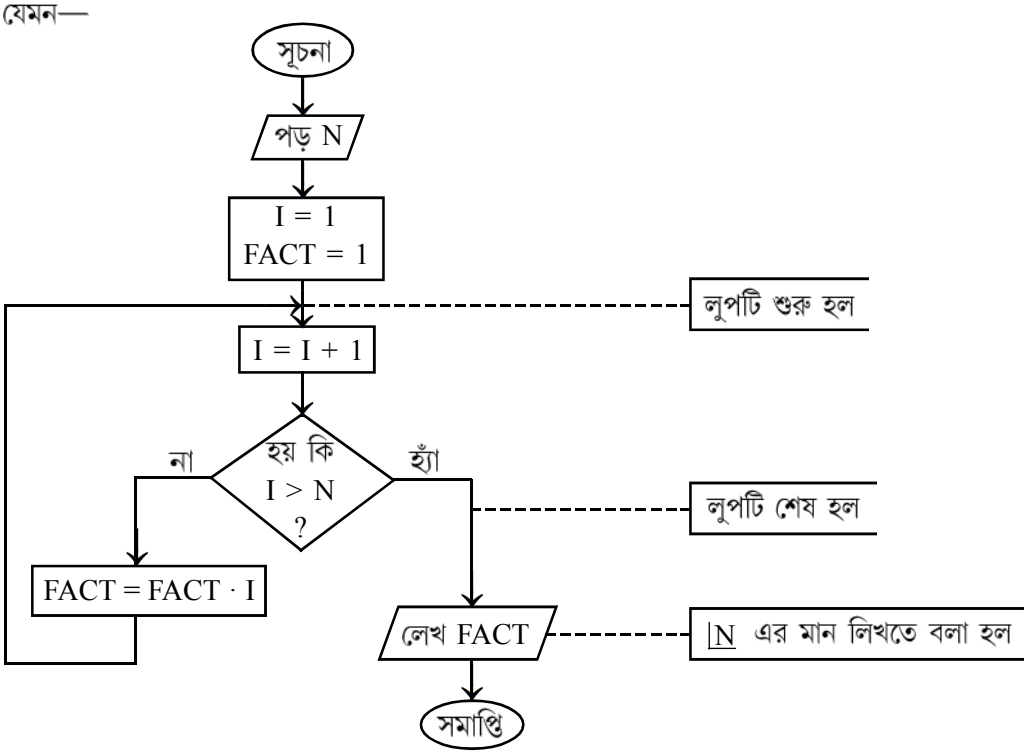
ক্রিয়া প্রবাহচিত্র অত্যন্ত সুদীর্ঘ হলে এবং একই পৃষ্ঠায় আঁকা সম্ভব না হলে প্রবাহচিত্রের কিছু অংশ অন্য পৃষ্ঠায় আঁকা যেতে পারে। এক্ষেত্রে ভিন্ন পৃষ্ঠাতে আঁকা অংশটিকে প্রারম্ভিক অংশের সাথে সম্পর্কযুক্ত করতে হবে দুটি বিশেষ আকৃতির পঞ্চভূজের সাহায্যে। পঞ্চভূজদুটি অবশ্যই সমচিহ্নযুক্ত হবে।

(7) মন্তব্য করার ক্ষেত্রচিহ্ন ঃ



(Comment/Annotation Box)

ক্রিয়া প্রবাহচিত্রের কোন জটিল অংশকে ব্যাখ্যা করতে অথবা অন্যদের অনুধাবনের সুবিধার্থে অনেক সময় কোন কোন ধাপের পাশে প্রয়োজনীয় মন্তব্যটি একটি একবাহু উন্মুক্ত আয়তাকার ক্ষেত্রচিহ্নের মধ্যে লেখা হয়। যেমন—

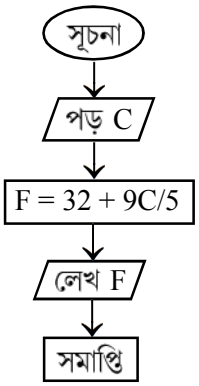


নীচে কয়েকটি ক্রিয়া প্রবাহচিত্রের উদাহরণ দেওয়া হল।

উদাহরণ-1 ঃ

সমস্যা ঃ সেল্‌সিয়াস স্কেলে প্রদত্ত উষ্ণতাকে ফারেনহাইট্ স্কেলে রূপান্তর করার জন্য একটি ক্রিয়াপ্রবাহ চিত্র আঁক।

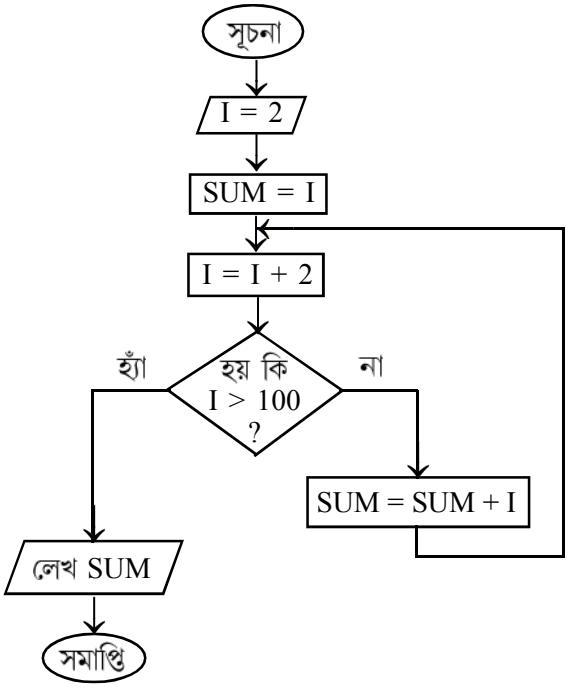
ক্রিয়া প্রবাহচিত্র ঃ



উদাহরণ-2 ঃ

সমস্যা ঃ 1 থেকে 100 এর মধ্যবর্তী যুগ্ম সংখ্যাগুলির সমষ্টি নির্ণয় করতে একটি ক্রিয়া প্রবাহচিত্র আঁক।

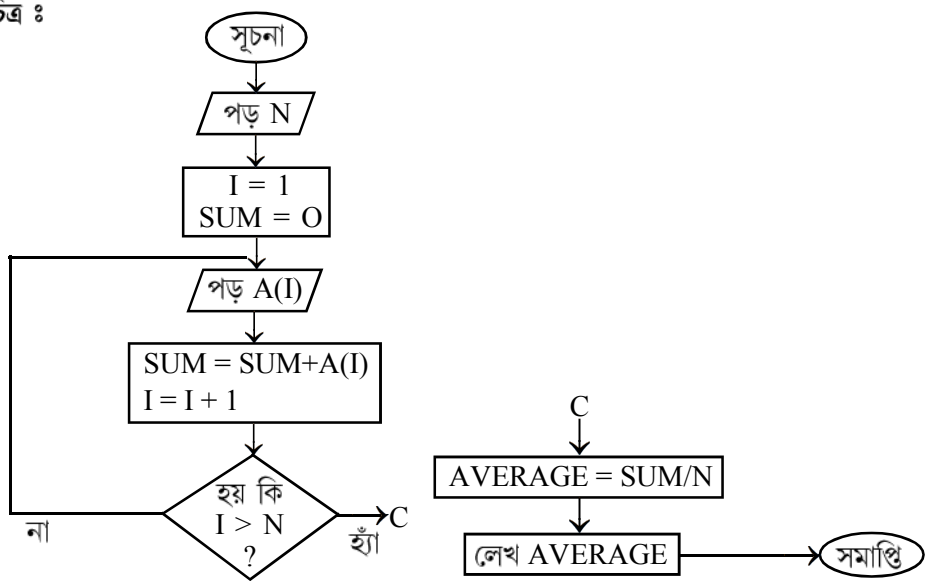
ক্রিয়া প্রবাহচিত্র ঃ



উদাহরণ-3 ঃ

সমস্যা ঃ N সংখ্যক প্রদত্ত সংখ্যার গড় নির্ণয় করার জন্য একটি ক্রিয়া প্রবাহচিত্র আঁক। সংখ্যাগুলি হল A(1), A(2), ..,A(N) দ্বারা নির্দেশিত।

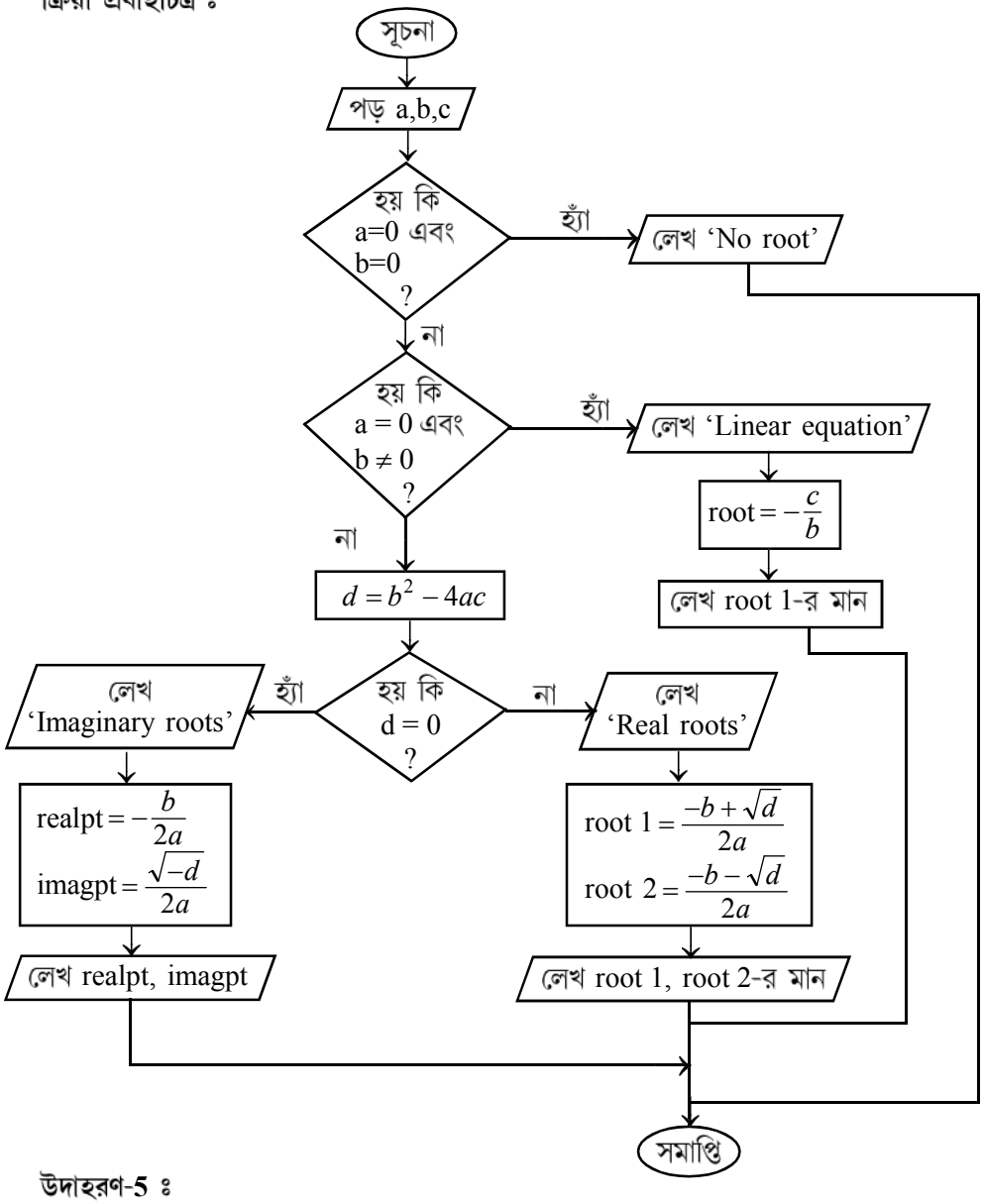
ক্রিয়া প্রবাহচিত্র ঃ



উদাহরণ-4 ঃ

সমস্যা ঃ $ax^2 + bx + c = 0$ এই দ্বিঘাত সমীকরণটি সমাধানের জন্য একটি ক্রিয়া প্রবাহচিত্র আঁক।

ক্রিয়া প্রবাহচিত্র ঃ



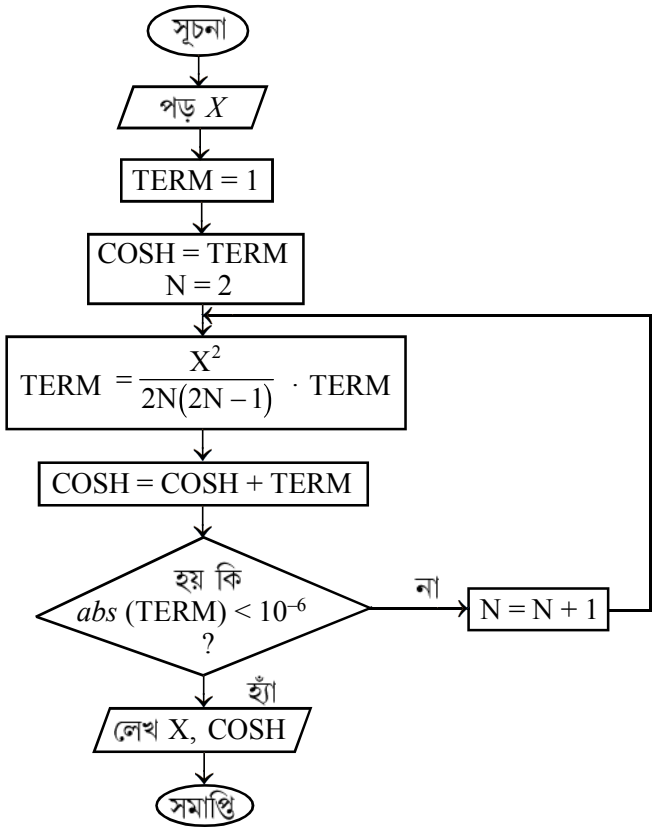
উদাহরণ-5 ঃ

সমস্যা ঃ $\cos h\, x$ এর মান পাঁচ দশমিক স্থান পর্যন্ত শুদ্ধ করে নির্ণয় করবার জন্য একটি প্রবাহচিত্ররেখা প্রস্তুত কর।

সমাধান ঃ আমরা জানি যে, $\cos h\, x = 1 + \frac{x^2}{\lfloor 2} + \frac{x^4}{\lfloor 4} + \frac{x^6}{\lfloor 6} + ...$

এই শ্রেণীটির n-তম পদ $= t_n = \frac{x^{2n-2}}{\lfloor 2n-2}$, n = 1, 2, 3, 4, ...

$\therefore \quad \frac{t_{n+1}}{t_n} = \frac{x^2}{2n(2n-1)}$ বা, $t_{n+1} = \frac{x^2}{2n(2n-1)} \cdot t_n$



উদাহরণ-6 ঃ

সমস্যা ঃ দ্বাদশ শ্রেণীর ছাত্রদের নাম, রোল নম্বর এবং ছয়টি বিভিন্ন বিষয়ে প্রাপ্ত নম্বর দেওয়া আছে। ছাত্রদের মোট নম্বর নির্ণয় কর এবং গড় নম্বরের ভিত্তিতে তাদের বিভাগ নির্ণয় কর। দেওয়া আছে।

60% ≤ গড় নম্বর	প্রথম বিভাগ
45% ≤ গড় নম্বর < 60%	দ্বিতীয় বিভাগ
35% ≤ গড় নম্বর < 45%	তৃতীয় বিভাগ
গড় নম্বর < 35%	অনুত্তীর্ণ

সংশ্লিষ্ট ক্রিয়াপ্রবাহটি এটি আঁক।

ক্রিয়া প্রবাহচিত্র ঃ

ধরা যাক, মোট ছাত্র সংখ্যা = N

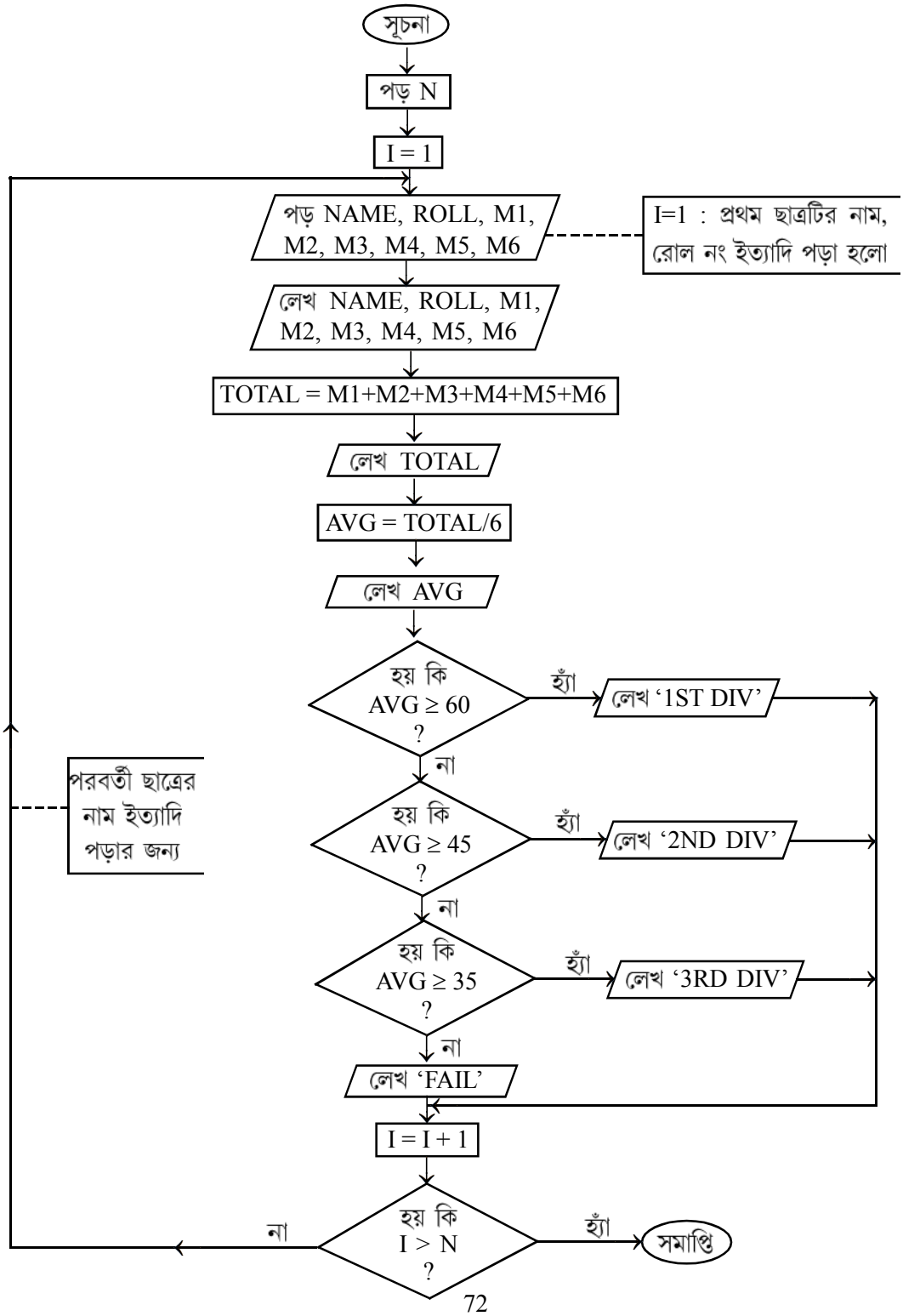
ছাত্রের নাম = NAME

ছাত্রের রোল নম্বর = ROLL

ছয়টি বিষয়ে প্রাপ্ত নম্বর = M_1, M_2, M_3, M_4, M_5, M_6

মোট নম্বর = TOTAL

গড় নম্বর = AVG


সূচনা
পড় N
I = 1
পড় NAME, ROLL, M1, M2, M3, M4, M5, M6
I=1 : প্রথম ছাত্রটির নাম, রোল নং ইত্যাদি পড়া হলো
লেখ NAME, ROLL, M1, M2, M3, M4, M5, M6
TOTAL = M1+M2+M3+M4+M5+M6
লেখ TOTAL
AVG = TOTAL/6
লেখ AVG
হয় কি AVG ≥ 60 ?
হ্যাঁ
লেখ '1ST DIV'
না
হয় কি AVG ≥ 45 ?
হ্যাঁ
লেখ '2ND DIV'
না
হয় কি AVG ≥ 35 ?
হ্যাঁ
লেখ '3RD DIV'
না
লেখ 'FAIL'
পরবর্তী ছাত্রের নাম ইত্যাদি পড়ার জন্য
I = I + 1
হয় কি I > N ?
না
হ্যাঁ
সমাপ্তি

সংক্ষিপ্তসার ঃ

পরিগণকের ভাষাতে কোন প্রোগ্রাম লিখতে হলে প্রথমে ক্রিয়ানুক্রম তালিকা তৈরী ও কর্মপ্রবাহ চিত্র অঙ্কন প্রয়োজন। এই এককটিতে ক্রিয়ানুক্রম তালিকা তৈরী করার পদ্ধতি বিভিন্ন উদাহরণ সহকারে আলোচনা করা হয়েছে। কর্মপ্রবাহ চিত্র অঙ্কন করার জন্য বিভিন্ন চিত্র ছোট ছোট উদাহরণসহ বর্ণনা করা হয়েছে। শেষে অনেকগুলো সমস্যার প্রয়োজনীয় সম্পূর্ণ কর্মপ্রবাহ চিত্র অঙ্কন করা হয়েছে।

একক 5 ❑ পরিগণকের ভাষা—C

এককের উদ্দেশ্য ঃ

এই একক পাঠ করলে পাঠক নিম্নলিখিত বিষয়গুলি সম্বন্ধে জানতে পারবে—

- পরিগণকের ভাষা কি?
- C ভাষার সৃষ্টির ইতিহাস
- C ভাষার ব্যবহৃত চিহ্নসমুহ
- C ভাষার বিশেষ অর্থবহ শব্দ
- বিভিন্ন ধরণের ধ্রুবক ও চলরাশি
- চলকের নামকরণ ও প্রকৃতি
- C ভাষার বিভিন্ন প্রক্রিয়া ও রাশিমালা
- অপেক্ষক ভাণ্ডার
- হেডার ফাইল ও তার ব্যবহার

পরিগণকের ভাষা—C

বর্তমানে প্রচলিত বিভিন্ন পরিগণকীয় ভাষাগুলির মধ্যে অন্যতম হলো C-ভাষাটি। UNIX অপারেটিং সিষ্টেমের সঙ্গে C-ভাষাটি বিশেষভাবে যুক্ত। আমেরিকার AT & T-এর Bell পরীক্ষাগারের প্রযুক্তিবিদ ও কম্প্যুটার বিশেষজ্ঞদের যৌথ প্রয়াসে ১৯৭২ সালে এই ভাষাটির সৃষ্টি। এর পূর্বে ১৯৬০-এর দশকে প্রথম দিকে যে সমস্ত পরিগণকীয় ভাষার প্রচলন ছিল সেগুলি হলো প্রধানত বৈজ্ঞানিক ও প্রযুক্তিবিদ্যায় সহায়ক, যেমন FORTRAN, BASIC ইত্যাদি। COBOL ভাষাটি ছিল ব্যবসা-বাণিজ্য সহায়ক ভাষা। এর সাহায্যে গাণিতিক সমস্যা সংক্রান্ত কোন গণনাকার্য সম্পন্ন করার কাজ ছিল বেশ কঠিন। এই কারণে এমন একটি ভাষার প্রয়োজন দেখা দিল যা কিনা বৈজ্ঞানিক/প্রযুক্তিবিদ্যা সংক্রান্ত গণনাকার্য্যে যেমন সহায়ক হবে, তেমনি ব্যবসা-বাণিজ্য সংক্রান্ত কাজেও হবে সমান সহায়ক। এই উদ্দেশ্য নিয়ে একটি আন্তর্জাতিক কমিটিও গঠিত হয়। এই কমিটি ALGOL-60 নামে একটি পরিগণকীয় ভাষার প্রস্তাবনা করে। তবে এই ভাষাটি যথেষ্ট ব্যবহার উপযোগী ছিল না। ইতিমধ্যে কেমব্রিজ বিশ্ববিদ্যালয়ের বিজ্ঞানী ও প্রযুক্তিবিদরা প্রায় একই উদ্দেশ্য নিয়ে একটি ভাষার প্রস্তাবনা করেন যার নাম দেওয়া হয় CPL (Combined Programming Language)—তবে এই ভাষাটিও ছিল যথেষ্ট দীর্ঘ এবং এর সাহায্যে কাজ করাও ছিল বেশ কঠিন। এর কিছু পরেই ঐ বিশ্ববিদ্যালয়ের মার্টিন রিচার্ডস্ CPL—ভাষাটিকে সংশোধিত ও পরিমার্জিত করে নতুন একটি ভাষার সৃষ্টি করেন—BCPL (Basic Computer Programming Language) ১৯৭০ সালে AT & T-এর কেন্ থম্প্সন্ (Ken Thompson) CPL ভাষাটির এক সরলীকৃত সংস্করণ প্রকাশ

করেন এবং ভাষাটির নাম দেন B—এই B—ভাষাটি CPL-এর প্রায় অনুরূপ। BCPL এবং B এই ভাষা দুটি CPL-এর তুলনায় সরলীকৃত হলেও সব ধরণের কাজের পক্ষে যথেষ্ট সক্ষম ছিল না—বিশেষ কিছু ক্ষেত্রেই এদের ব্যবহার ছিল সীমাবদ্ধ। অবশেষে ১৯৭২ সালে Bell—ল্যাবোরেটরীর ডেনিস্ রিচি (Dennis Ritchie) এই দুটি ভাষাকে একত্রিত করেন এবং সংশোধিত ও পরিমার্জিত রূপে নতুন আর একটি ভাষার প্রবর্তন করেন—এই ভাষাটিকেই বলা হল C-পরিগণকীয় ভাষা। পরবর্তী কালে এই ভাষাটিকে নানা স্থানে নানা ভাবে পরিবর্তন ও সংশোধন করা হতে থাকে। অসংগঠিত ভাবে এই প্রকার পরিবর্তনের ফলে C-ভাষাটির একাধিক ভিন্ন ভিন্ন রূপ প্রচলিত হয়। এই বিভ্রান্তি নিরসনের জন্য ১৯৮৩ সালে ANSI কর্তৃক একটি কমিটি গঠন করা হয়। ঐ কমিটি C-ভাষাটির একটি প্রামাণ্য রূপ প্রকাশ করেন। এই প্রামাণ্য রূপটিকেই বলা হয় ANSI C। ANSI C, UNIX অথবা MSDOS যে কোন অপারেটিং সিস্টেমেই বিশেষভাবে কার্যকরী। এই ভাষাটি অত্যন্ত কার্যকরী এবং অন্যান্য সমকালীন ভাষা যেমন BASIC, FORTRAN—প্রভৃতির তুলনায় অনেক দ্রুত। কোন কোন ক্ষেত্রে BASIC -এর তুলনায় এটি প্রায় ৫০ গুণ বেশী দ্রুত। এর কার্যকারীতা এবং সক্ষমতার মূল কারণ হলো এটি একটি অপেক্ষক-নির্ভর (Functions-oriented) ভাষা, যার বিভিন্ন উপাংশ অপেক্ষকগুলি একযোগে ভিন্ন ভিন্ন প্রকার কাজ সম্পন্ন করে। এর ফলে সমগ্র কাজটি অনেক কম সময়ে সুসম্পন্ন হয়। C-ভাষাটি প্রধানতঃ ব্যবহারকারীর উপযোগী একটি উচ্চস্তরের ভাষা হলেও পরিগণক যন্ত্রের ও (যা প্রধানতঃ নিম্নস্তরের ভাষার উপযোগী) পক্ষে যথেষ্ট সহায়ক। এই দ্বৈত সত্ত্বার জন্য অনেক সময় C-ভাষাকে মধ্যস্তরীর ভাষা হিসাবে মনে করা হয়।

যে কোন কথোপকথনের ভাষা (যেমন বাংলা, ইংরাজী ইত্যাদি) শিখতে আমরা প্রথমেই তার বর্ণমালা, এবং বিভিন্ন যতি চিহ্ন, বর্ণমালার সংযোগে গঠিত পদ, পদের সংযোগে গঠিত বাক্য ইত্যাদির সঙ্গে পরিচিত হই। সঠিক বাক্য গঠনের জন্য শিখতে হয় ঐ ভাষার ব্যাকরণ। অনুরূপভাবে পরিগণকীয় ভাষার শিক্ষার জন্যও এই একই কথাই প্রযোজ্য। C-ভাষাটি শিখতে হলে প্রথমেই জানতে হবে এই ভাষার ব্যবহৃত বর্ণমালা ও অন্যান্য চিহ্নগুলি কি কি।

C-বর্ণমালা ও অন্যান্য চিহ্ন (Character Set) : এই সেট্-এর মধ্যে আছে বর্ণমালা বা Alphabet, দশমিক সংখ্যা, যতিচিহ্ন এবং অন্যান্য বিশেষ চিহ্নসমূহ।

বর্ণমালা ঃ A, B, C, ..., X, Y, Z (বড় হাতের ইংরাজী বর্ণমালা মোট ২৬টি)

a, b, c, ..., x, y, z (ছোট হাতের ইংরাজী বর্ণমালা মোট ২৬ টি)

দশমিক সংখ্যা ঃ 0, 1, 2, ..., 8, 9 (মোট ১০টি)

যতিচিহ্ন ও অন্যান্য চিহ্ন সমূহ ঃ

, (কমা)

; (সেমিকোলন)

. (দশমিক বিন্দু)

? (জিজ্ঞাসা চিহ্ন)

’ (লুপ্ত কমা অথবা কোটেশন্ চিহ্ন)

: (কোলন)

! (আশ্চর্যবোধক চিহ্ন)

” (ডবল্ কোটেশন চিহ্ন)

| (লম্ব দণ্ড)

/ (তির্যক রেখা)

\ (উল্টা তির্যক রেখা)

+ (যোগ চিহ্ন)

– (বিয়োগ চিহ্ন বা হাইফেন্)

* (তারকা চিহ্ন/Asterisk)

& (‘এবং’ অর্থসূচক সংকেত/Ampersand)

= (সমান চিহ্ন)

$ (ডলার চিহ্ন)

% (শতকরা চিহ্ন)

(Hash-চিহ্ন)

^ (ক্যারেট/Caret)

((বাম লঘু বন্ধনী)

) (ডান লঘু বন্ধনী)

[(বাম গুরুবন্ধনী)

] (ডান গুরুবন্ধনী)

{ (বাম ধনুর্বন্ধনী/Left brace)

} (ডান ধনুর্বন্ধনী/Right brace)

< (বাম কৌণিক বন্ধনী/ক্ষুদ্রতর চিহ্ন)

> (ডান কৌণিক বন্ধনী/বৃহত্তর চিহ্ন)

~ (টিলড্ চিহ্ন/Spanish ভাষার ব্যবহৃত হয় এমন (Tilde) একটি চিহ্ন)

__ (নিম্নরেখা চিহ্ন/Underscore)

C–ভাষার বিশেষ অর্থবহ কিছু শব্দ (Key Word) ঃ

পরিগনকীয় ভাষায় বর্ণমালার সমন্বয়ে তৈরী হয় শব্দ, শব্দগুলি বিশেষ অর্থ নির্দেশ করে। যেমন FORTRAN ভাষায় কোন চলকের মান পড়বার জন্য ব্যবহার করা হয় READ শব্দটি, লেখার জন্য WRITE ইত্যাদি। C-ভাষাতেও অনুরূপ কিছু আছে যা কেবলমাত্র বিশেষ বিশেষ অর্থেই ব্যবহৃত হয়। ANSI C-তে এই

ধরণের ৩২ শব্দ আছে। এদের মধ্যে বহুল ব্যবহৃত উল্লেখযোগ্য কয়েকটি শব্দ হলো—auto break case char continue default do double else extern float for goto if int long return static short signed switch typedef void while ইত্যাদি।

ধ্রুবক ও চলরাশি ঃ সাধারণ বীজগণিতে (অর্থাৎ বাস্তব সংখ্যার বীজগণিত) আমরা প্রায়শই $3x - 8y = 3{\cdot}4$ এই ধরণের বীজগাণিতিক রাশি বা সমীকরণের দেখা পাই। এই সমীকরণে 3 এবং 3·4 হলো ধ্রুবক এবং x ও y হলো চলরাশি, যাদের মান পরিবর্তনশীল। উপরের সমীকরণে তিনটি ধ্রুবক এবং দুটি চলরাশি আছে। C-পরিগণকীয় ভাষাতেও অনুরূপভাবে ধ্রুবক ও চলরাশির ব্যবহার আছে।

C–ধ্রুবক ঃ সাধারণ গণিতশাস্ত্রের নিয়ম অনুযায়ী ধ্রুবক হলো এমন একটি রাশি যা পরিবর্তনশীল নয়, অর্থাৎ কোন গাণিতিক কার্য চলাকালীন এগুলির মান একই থাকে। C–ভাষায় তিন ধরণের ধ্রুবকের ব্যবহার আছে—সাংখ্যিক (অর্থাৎ যে সকল ধ্রুবকের মান 3, –1·5, 0·27, 12 বা অনুরূপ কোন সাংখ্যিক মান) বর্ণ এবং তন্ত্রী ধ্রুবক এই ধরণের ধ্রুবকের মান সাংখ্যিক নয়, কোন ব্যক্তি, বস্তু, ঠিকানা বা অনুরূপ কোন 'মান' হতে পারে)

C–ধ্রুবকের প্রকারভেদ ও উদাহরণ ঃ

(১) সাংখ্যিক ধ্রুবক ঃ এই ধ্রুবকের মান সাংখ্যিক অর্থাৎ এই ধরণের ধ্রুবকের মান –123·05, 2, 3, –8, 17·12, –12·0, 12, 12·0 বা এই প্রকার কোন সংখ্যা হতে পারে। এদেরকে দুভাগে ভাগ করা যেতে পারে—পূর্ণ সাংখ্যাক (যেমন 2, 3, –12, 0, +48 ইত্যাদি অর্থাৎ দশমিক বিন্দুবিহীন কোন সংখ্যা) এবং বাস্তব (যেমন 2·0, –12·12, 18·9, –4·0, 0·0 ইত্যাদি—অর্থাৎ দশমিক বিন্দু যুক্ত কোন সংখ্যা)। এখানে উল্লেখ করা প্রয়োজন যে বাস্তব ধ্রুবকের ক্ষেত্রে 'বাস্তব' শব্দটি কেবলমাত্র দশমিক বিন্দুর উপস্থিতিই নির্দেশ করে, অন্য কোন গাণিতিক ব্যাখ্যা নেই।

পূর্ণসাংখ্যিক ধ্রুবকের ক্ষেত্রে কয়েকটি নিয়ম মনে রাখা দরকার ঃ

(i) এই ধ্রুবকগুলির মান পূর্ণসংখ্যা হতে হবে। 0, 1, 2, ..., 8, 9 কেবলমাত্র এই অঙ্কগুলির সাহায্যেই ধ্রুবকের মান লিখতে হবে, অন্য কোন চিহ্ন ব্যবহার করা যাবে না। ধনাত্মক ধ্রুবকের ক্ষেত্রে শুরুতেই + চিহ্নটি ব্যবহার করা যেতেও পারে তবে ঋণাত্মক ধ্রুবকের ক্ষেত্রে প্রথমেই — চিহ্নটি থাকতে হবে।

(ii) ধ্রুবকের মানে কোন দশমিক বিন্দু থাকবে না। দশমিক বিন্দু থাকলে ধ্রুবকটিকে 'বাস্তব' ধ্রুবক বলা হবে।

(iii) এক অঙ্ক বিশিষ্ট ধ্রুবকের ক্ষেত্রে, যেমন 5, তাকে 05 লেখা যাবে না, শুধুমাত্র 5 লিখতে হবে।

(iv) বহু অঙ্ক বিশিষ্ট ধ্রুবকের ক্ষেত্রে পরপর দুটি অঙ্কের মাঝে শূন্যস্থান না থাকাই বাঞ্ছনীয়, থাকলেও অবশ্য তার কোন প্রভাব নেই।

সঠিক বা সঠিক নয় এমন কতকগুলি পূর্ণসাংখ্যিক ধ্রুবকের উদাহরণ ঃ

123 (সঠিক)

12·3 (সঠিক নয়, দশমিক বিন্দু থাকার দরুণ এটি একটি বাস্তব ধ্রুবক)

46,203 (সঠিক নয়, কারণ কমা (,) চিহ্নটি আছে, অবশ্য সাধারণ ভাষাতে আমরা প্রায়শই এভাবে লিখে থাকি, কিন্তু C-ভাষাতে এটি চলবে না)

489 + (সঠিক নয়, + বা — চিহ্নটি সংখ্যার পূর্বে বসবে, পরে বা মাঝখানে থাকবে না)

37 139 (সঠিক, তবে 7 ও 1-এর মাঝে শূন্যস্থানটি না থাকাই বাঞ্ছনীয়)

Rs. 5000 (সঠিক নয়, Rs, $, £ ইত্যাদি চিহ্ন থাকবে না)

C–ভাষাতে দশমিক ধ্রুবক ছাড়াও অষ্টনিধানী (octal) বা ষোড়শ নিধানী (hcxadecimal) পদ্ধতিতেও ধ্রুবকের ব্যবহার আছে। অষ্টনিধানী ধ্রবকের প্রথম অঙ্কটি O দিয়ে শুরু হবে এবং ষোড়শ নিধানীর ক্ষেত্রে OX (অথবা Ox) দিয়ে শুরু করতে হবে। ধরা যাক দশমিক ধ্রুবকটি হল 69, অষ্টনিধানী পদ্ধতিতে এটি হল 105 এবং C–ভাষায় লিখতে হবে 0105। আবার ষোড়শ নিধানী পদ্ধতিতে দশমিক সংখ্যা 59-এর সমতুল্য সংখ্যাটি হলো 3B এবং C–ভাষাতে এটিকে লিখতে OX3B (অথবা OX3b অথবা Ox3B অথবা Ox3b)। অবশ্য এই ধরণের ধ্রুবকের ব্যবহার সাধারণত খুবই সীমিত। পূর্ণসাংখ্যিক ধ্রুবকের পরিসর নির্ভর করে পরিগণকের শব্দ-দৈর্ঘ্যের উপর। ধরা যাক্ একটি 16–বিট্ পরিগণক নেওয়া হল, অর্থাৎ এই পরিগণকের শব্দের বিট সংখ্যা 16 (অর্থাৎ 2 বাইট)। এই পরিগণকের স্মৃতিতে চিহ্ন বিহীন (অর্থাৎ ধনাত্মক) সর্ব্বোচ্চ যে সংখ্যাটিকে ধরে রাখা যাবে সেটি হলো 1111 ..... 11 (16-টি 1) (দ্বিনিধানী পদ্ধতিতে)—দশমিক পদ্ধতিতে যার মান হলো $1\times2^{15}+1\times2^{14}+1\times2^{13}+.....+1\times2^{2}+1\times2^{1}+1\times2^{\circ}$ $=1+2+2^{2}+2^{3}+.....+2^{14}+2^{15}=2^{16}-1$ (গুণোত্তর প্রগতির যোগফল) = 65535। সর্বনিম্ন সংখ্যাটি অবশ্যই 000 ..... 00 (16টি 0) = 0। অর্থাৎ শুধুমাত্র ধনাত্মক সংখ্যার কথা ধরলে ঐ পরিগণকে 0 থেকে $2^{16}-1$ (= 65535) এই পরিসরের মধ্যে অবস্থিত যে কোন ধ্রুবকের মান সঞ্চিত করা যাবে। এই প্রকার ধ্রুবকদের unsigned short integer (বা unsigned integer) হিসাবে চিহ্নিত করা হয়। তবে, ধনাত্মক বা ঋণাত্মক উভয় প্রকার ধ্রুবকের ক্ষেত্রে এই পরিসরটি হবে -2^{15} থেকে $2^{15}-1$ অর্থাৎ – 32768 থেকে 32767 পর্যন্ত। এই প্রকার ধ্রুবকদের signed short integer (বা short integer) বলা হয়। এর থেকে বৃহত্তর কোন পূর্ণসাংখ্যিক ধ্রুবকের প্রয়োজন হলে আমরা long integer-এর কথা ভাবতে পারি। কোন ধ্রুবককে long integer হিসাবে চিহ্নিত করলে পরিগণক আগের তুলনায় দ্বিগুণ পরিমান স্থান নির্দ্দিষ্ট করে রাখে ঐ ধরণের ধ্রুবকের মান সঞ্চিত করার জন্য। অর্থাৎ 16-bit এর কোন পরিগণককে ঐ ধরনের 'Long integer'-কে ধরে রাখার জন্য 32টি bit ব্যবহার করা হয়। অর্থাৎ long integer-এর ক্ষেত্রে পরিসরটি হলো -2^{31} থেকে $2^{31}-1$ (অর্থাৎ— 21, 47, 483, 648 থেকে + 21, 47, 483, 647 পর্যন্ত)। তবে ধ্রুবকটি চিহ্নবিহীন ধনাত্মক হলে এই পরিসরটি হয় 0 থেকে $2^{32}-1$ অর্থাৎ 0 থেকে 4294, 96, 72, 965 পর্যন্ত।

উপরের এই আলোচনা থেকে একটি ব্যাপার বেশ পরিস্কার যে পূর্ণসাংখ্যিক ধ্রুবকও বিভিন্ন আকারের হতে পারে, যেমন unsigned short, signed short, unsigned long বা signed long ইত্যাদি।

পূর্ণসাংখ্যিক ধ্রুবকের ব্যবহার সীমিত, কারণ গাণিতিক কার্য্যে অধিকাংশ ক্ষেত্রেই আমরা বাস্তব অর্থাৎ দশমিকযুক্ত ধ্রুবকের ব্যবহার করে থাকি। এই ধরনের ধ্রুবকের উদাহরণ আগেই দেওয়া হয়েছে— যেমন – 29·14, 12·0, 3·14159, 0·0008, + 1·10, – 0·18, 13· অথবা ·25 ইত্যাদি। এই সকল ধ্রুবকদের সাধারণতঃ বাস্তব (বা Floating Point) ধ্রুবক হিসাবে চিহ্নিত করা হয়। দশমিক চিহ্নযুক্ত এই প্রকার ধ্রুবকদের লিখিত আকারে প্রকাশ করার দুইটি পদ্ধতি আছে—ভগ্নাংশ আকারে এবং সূচক আকারে। **উদাহরণঃ** ভগ্নাংশ আকারে 13·29, – 3·14, ·32, 0·0073 ইত্যাদি। এখানে দশমিক বিন্দুর বামদিকের অংশটি পূর্ণঅংশ, এবং দশমিক বিন্দুর ডান দিকের অংশটি ভগ্নাংশ।

সূচক আকারে এই ধ্রুবকগুলিকে লেখা হয় যথাক্রমে 0·1329e02, – 0·314e01, 0·32e00, 0·73e-02। এখানে e02-র অর্থ হলো 10^2, e – 02 হলো 10^{-2} ইত্যাদি। অর্থাৎ $0 \cdot 1329e02 = 0 \cdot 1329 \times 10^2 = 13 \cdot 29$, এখানে 0·1329 কে বলা হয় Mantissa-অংশ এবং 02 হলো সূচক অংশ (বা exponent)।

সাধারণভাবে সূচকের মান ধনাত্মক বা ঋণাত্মক পূর্ণসংখ্যা হয় এবং এর Mantissa -অংশটি বাস্তব বা ভগ্নাংশ আকারে প্রকাশ করা হয়। এই Mantissa অংশটি সাধারণক্ষেত্রে ৮ সার্থক অঙ্ক পর্যন্ত শুদ্ধ হয়ে থাকে তবে বিশেষক্ষেত্রে একে ১৬ সার্থক স্থান পর্যন্ত শুদ্ধ করা যেতে পারে, তবে এর জন্য প্রথমেই পরিগণককে প্রয়োজনীয় নির্দেশ দেওয়া প্রয়োজন। কোন বাস্তব ধ্রুবক সাধারণভাবে পরিগণকের স্মৃতির 4-বাইট স্থান অধিকার করে থাকে।

(২) বর্ণ ধ্রুবক ঃ এই ধরণের ধ্রুবকগুলিকে লেখা হয় 'A', '3', '*', 'd', ' , ' , ' ৳ ' —এই প্রকারে। অর্থাৎ দুটি quote (একমাত্রার) চিহ্নে দ্বারা আবদ্ধ একটি মাত্র বর্ণ, সংখ্যা বা অন্য কোন C-চিহ্ন। এই ধ্রুবকগুলির সাংখ্যিক মান হলো ঐ চিহ্নটির ASCII -রূপের সমতুল্য দশমিক মান। যেমন

'A' এই ধ্রুবকটির সাংখ্য মান হলো $(01000001)_2 = (65)_{10}$ কারণ ASCII-কোড-এ A-র কোড হলো 01000001। আরো কয়েকটি চারিত্রিক ধ্রুবক ও তার সাংখ্য মান নীচে দেওয়া হলো ঃ

বর্ণ ধ্রুবক	দশমিক সাংখ্যা মান
'5'	53
' √ '	251
' ৳ '	255 (৳ শূন্যস্থান নর্দেশ করে)
' = '	61

যে সকল পরিগণক যন্ত্রে ASCII কোডের পরিবর্তে EBCDIC কোড ব্যবহৃত হয় (প্রধানত IBM-প্রযুক্তি নির্ভর যন্ত্র) সেক্ষেত্রে এই সাংখ্য মানগুলি EBCDIC-কোড অনুযায়ী হবে—যেমন 'A'-র সাংখ্যমান EBCDIC কোড অনুয়ায়ী হবে 193, 65 নয়। এই ধরণের বর্ণ ধ্রুবকগুলি গাণিতিক প্রক্রিয়ায় ব্যবহার করা যেতে পারে। উদাহরণস্বরূপ মনে করা যাক X = 'A' এবং Y = '?' সেক্ষেত্রে X – Y = 'A' – '?' = 65 – 63 = 2।

তন্ত্রী ধ্রুবক (String Constants) ঃ তন্ত্রী ধ্রুবক সাংখ্যিক নয়, এর কোন সাংখ্যমান নেই। একজোড়া ডবল quote—চিহ্নের মধ্যে যে কোন বর্ণ, চিহ্ন, সংখ্যা ইত্যাদি পরপর লিখলে সেটি একটি তন্ত্রী ধ্রুবক হিসাবে পরিগণিত হয়। যেমন ঃ

" NETAJI ৳ SUBHAS ৳ UNIVERSITY"

" A"

" NAME = A. B. ROY"

" 4 + 7 = 19"

" * + ? = /A3"

" Address : 32, Amherst Street, Kolkata-700009"

এখানে উল্লেখ্য যে বর্ণ ধ্রুবক 'A' এবং তন্ত্রী ধ্রুবক "A" দুটি সম্পূর্ণ পৃথক ধ্রুবক। প্রথমটির একটি সাংখ্যমান

আছে, এবং গাণিতিক গণনাকার্য্যে ব্যবহৃত হতে পারে; কিন্তু দ্বিতীয়টির কোন সাংখ্যমান নেই, তবে বিভিন্ন শিরোনাম, মন্তব্য ইত্যাদি লেখার কাজে এ ধরণের তন্ত্রী ধ্রুবক ব্যবহৃত হয়।

C–পরিগণকীয় ভাষার চলক ও তার ব্যবহার

কোন গাণিতিক গণনাকার্য্যে চলকের মান পরিবর্তনশীল, গণনাকার্য্যের বিভিন্নস্তরের এই চলকগুলির ভিন্ন ভিন্ন মান হতে পারে। সাধারণ বীজগণিতের মত এই চলকগুলিরও নামকরণ দরকার। সাধারণভাবে আমরা চলক হিসাবে x,y,z ইত্যাদি চিহ্ন ব্যবহার করে থাকি। C-ভাষাতেও চলকগুলির নাম হিসাবে অনুরূপ চিহ্ন ব্যবহার করা যেতে পারে।তবে এক্ষেত্রে নামকরণের কিছু বিধিনিষেধ আছে, যেগুলি মেনে চলতে হবে। ANSI C ভাষাতে এই ধরণের বিধিনিষেধগুলি হলো ঃ

(১) চলকের নামের প্রথম অক্ষরটি অবশ্যই কোন বর্ণমালা, যেমন a, b, c, ... ইত্যাদি হতে হবে। পরবর্তী অক্ষরগুলি বর্ণ বা সংখ্যা বা—(নিম্নরেখা/underscore) হতে পারে।

(২) মোট অক্ষরসংখ্যা সাধারণভাবে আট বা তার কম হতে হবে, তবে বর্তমানে অক্ষরসংখ্যা সর্বোচ্চ 31 পর্যন্ত হতে পারে (কোন কোন মেশিনে 40 পর্যন্তও হতে পারে)

(৩) নামের মধ্যে উপরোক্ত চিহ্নগুলি ছাড়া অন্য কোন চিহ্ন ব্যবহার করা যাবে না। শূন্যস্থান সাধারণভাবে গ্রাহ্য হয় না।

(৪) নামকরণের ক্ষেত্রে C–ভাষার বিশেষ শব্দগুলি ব্যবহার না করাই বাঞ্ছনীয়।

(৫) বড় হাতের বা ছোটহাতের সব প্রকার বর্ণই ব্যবহার করা যেতে পারে, তবে সেক্ষেত্রে চলকগুলি ভিন্ন ভিন্ন চলক হবে। যেমন MASS, mASS mass, Mass এই চলকগুলি পৃথক পৃথক চলক। তবে সাধারণভাবে C–ভাষাতে ছোটহাতের বর্ণগুলিই বেশী ব্যবহার করা হয়।

কতকগুলি বৈধ চলকের নাম ঃ

a, aa, a1, a11, mass, v123, n_s_o_u, int_var, (int শব্দটি বিশেষশব্দ হলেও, এক্ষেত্রে এটি ব্যবহার করা হয়েছে চলকের নামে অংশ-বিশেষ হিসাবে)

নীচের নামগুলি চলকের বৈধ নাম নয় ঃ

4 mass (প্রথম অক্ষরটি বর্ণ নয়)

goto (বিশেষ শব্দ/যদিও বর্তমানে এই বিশেষ শব্দগুলি চলকের নামকরণে ব্যবহার করার বিধিনিষেধ অনেকটা শিথিল করা হয়েছে)

Vol, den (বিশেষ চিহ্ন থাকবে না)

চলকের প্রকৃতি

C–ভাষাতে কোন চলক তার নামদ্বারা চিহ্নিত। পরিগণক তার স্মৃতিতে চলকগুলির মান ধরে রাখার জন্য এক বা একাধিক বাইট্ সংরক্ষিত করে রাখে। এই চলকগুলির প্রকৃতি তার মানের প্রকৃতির সঙ্গে সঙ্গতিপূর্ণ হতে হবে, অর্থাৎ কোন চলকের মান যদি শুধুমাত্র পূর্ণসংখ্যার হয় তবে চলকটিতে পূর্ণসাংখ্যিক চলক বলা হবে। অনুরূপভাবে কোন চলক বাস্তব (বা floating point) চলক, বর্ণ চলক বা তন্ত্রীচলকও হতে পারে।

C–ভাষাতে চলকের প্রকৃতি নিরূপক নির্দেশ

কোন চলক যদি পূর্ণসাংখ্যিক হয়, তবে পরিগণক সাধারণভাবে তার মান স্মৃতিতে ধরে রাখার জন্য 2-বাইট পরিমান স্থান সংরক্ষিত করে রাখে। long integer-এর ক্ষেত্রে 4-বাইট্, বাস্তব চলকের জন্য 4-বাইট, Double Floating-এর 8 বা 10-বাইট্ স্থান সংরক্ষিত হয়। অতএব পরিণকের সাহায্যে কোন গাণিতিক কাজ শুরু করার প্রথমেই সংশ্লিষ্ট চলক্গুলির প্রকৃতি নির্দেশ করে দেওয়া আবশ্যিক। এতে গণনার কাজ অনেক দ্রুততর হবে এবং পরিগণকও বিভান্তি মুক্ত হবে এবং নির্দেশ অনুসারে তার স্মৃতিতে যথোপযুক্ত স্থান সংরক্ষন করে রাখবে।

এই প্রকৃতি নিরুপক নির্দেশগুলি নীচে উল্লেখ করা হলো ঃ

তিনটি C-চলক নেওয়া হল যাদের নামগুলি হলো mass, roll এবং area

(১) তিনটি চলকই চিহ্নযুক্ত পূর্ণসাংখ্যিক চলক হলে, নির্দেশটি হবে ঃ

```
int mass;
int roll;
int area;
```

অথবা `int mass, roll, area;`

(২)
```
short int mass;
      int roll;
long  int area;
```

এই নির্দেশের সাহায্যে mass-এই চলককে short integer হিসাবে, roll-এই চলকটিকে integer হিসাবে এবং চলক area-কে long integer হিসাবে চিহ্নিত করা হয়।

(৩) ধরা যাক্ mass-এই চলকটি বাস্তব (floating point) চলক, roll এই চলকটি double floating (অর্থাৎ 16 সার্থক অঙ্ক পর্যন্ত সঠিক) এবং area চলকটি বর্ণ চলক (character & variable), এবং চতুর্থ আর একটি চালক, count হলো পূর্ণসাংখ্যিক চালক। অর্থাৎ mass-চলকটির মান দশমিক বিন্দুযুক্ত এবং এটি সর্ব্বোচ্চ 8 সার্থক অঙ্ক পর্যন্ত নির্ভুল, roll চলকটি মানও দশমিক বিন্দুযুক্ত তবে এটি সর্বোচ্চ 16 টি সার্থক অঙ্ক পর্যন্ত নির্ভুল। area চলকটি বর্ণচলক অর্থাৎ এর মান 'A', '5', '$\sqrt{}$' এই প্রকার হতে পারে, যার সাংখ্যিক মান বর্ণটির ASCII-মানের সমান। count চলকটি পূর্ণসাংখ্যিক অর্থাৎ এর সাংখ্যমান – 4, 3, 0, – 0, 12 এই প্রকারের হতে পারে। পরিগণককে এক্ষেত্রে যে নির্দেশ দিতে হবে তা হলো ঃ

```
float mass;
double roll;
```

char area;

int count;

float, int, double, char এই শব্দগুলি হলো C–ভাষায় চলকের প্রকৃতি নির্দেশক শব্দ।

(৪) তন্ত্রী (string) চলকের ক্ষেত্রে যে নির্দেশক শব্দটি ব্যবহার করা হয় সেটি হলো char, তবে তার ব্যবহার বর্ণচলকের চেয়ে সামান্য পৃথক। মনে করা যাক mass হলো এমন একটি তন্ত্রী চলক যার সর্বোচ্চ 20টি পর্যন্ত বর্ণ হতে পারে। সেক্ষেত্রে এই চলকটির প্রকৃতি ঘোষণাকারী নির্দেশটি হল ঃ

char mass [20]

এই নির্দেশটির সাধারণরূপ হলো ঃ

char a[m], b [n];

যেখানে a এবং b হলো তন্ত্রীচলক এবং m ও n হলো ধনাত্মক পূর্ণসংখ্যা যা ওই চলকদুটির সর্বোচ্চ বর্ণ সংখ্যা, অর্থাৎ চলকদুটির size (শূন্যস্থান ৳ একটি বর্ণ হিসাবে পরিগণিত হবে)

চলকের প্রকৃতি নির্দেশ ও প্রাথমিক মান প্রদান

উপরের নির্দেশগুলি কোন চলকের প্রকৃতি ঘোষণা বা নির্দেশ করে। আগেই বলা হয়ে চলকের মান পরিবর্তনশীল। কোন গণনাকার্য্যে কোন চলক প্রথম যে মানটি গ্রহণ করে বা প্রথম যে মানটি দিয়ে গণনার কাজ শুরু করা হয় সেটি হলো ঐ চলকের প্রাথমিক মান (initial value)। চলকের প্রকৃতি ঘোষণার সাথে সাথে তার প্রাথমিক মানটিও নির্দ্দিষ্ট করে দেওয়া যেতে পারে। এর জন্য নীচের নির্দেশগুলি ব্যবহার করা হয় ঃ

int mass = 15;

float area = 20·35;

char roll = 'M';

short int length = 3;

char name [ ] = "NETAJI ৳ SUBHAS";

শেষের উদাহরণটিতে [ ]-এর মধ্যে তন্ত্রীচলকটির সাইজ নির্দেশক ধনাত্মক কোন পূর্ণসংখ্যা লেখা হয়নি, এক্ষেত্রে পরিগণক নিজে থেকেই এই চলকটির একটি সাইজ নির্ধরণ করে নেবে যা হবে চলকটির সাইজ (অর্থাৎ বর্ণসংখ্যা) + 1 এক্ষেত্রে name–চলকটির বর্ণসংখ্যা হলো 13 (একটি শূন্যস্থান সমেত), অতএব name–এই চলকটির সর্বোচ্চ বর্ণসংখ্যা বা সাইজ ধরা হবে 13+1=14

C–ভাষায় বিভিন্ন প্রক্রিয়া ও রাশিমালা এবং সমীকরণ

C–ভাষায় বিভিন্ন চলক ও ধ্রুবক, বিভিন্ন প্রক্রিয়ার সাহায্যে যুক্ত করে C–রাশিমালা এবং সমীকরণ তৈরী করা হয়। এই প্রক্রিয়াগুলি গাণিতিক, যৌক্তিক অথবা সম্পর্কিত প্রক্রিয়া হতে পারে। বস্তুতপক্ষে, C–ভাষায় মোট ছয় প্রকারের প্রক্রিয়ার কথা বলা যেতে পারে। সেগুলি হল ঃ

(১) গাণিতিক প্রক্রিয়া (Arithmetic operatians)

(২) সম্পর্কিত (বা তুলনামূলক) প্রক্রিয়া (Relational operation)

(৩) যৌক্তিক প্রক্রিয়া (Logical operation)

(৪) মান প্রদানকারী প্রক্রিয়া (Assignment operation)

(৫) চলকের মান বৃদ্ধিকরণ ও হ্রাসকরণ প্রক্রিয়া (Increment and decrement operation)

(৬) শর্তাধীন প্রক্রিয়া (Conditional operation)

(১) গাণিতিক প্রক্রিয়া ঃ

C–ভাষায় গাণিতিক প্রক্রিয়াগুলি হলো যোগ, বিয়োগ, গুণ, ভাগ এবং মডুলো ভাগ (Modulo Division)। দুইটি চলকের মধ্যে এই প্রক্রিয়াগুলির মাধ্যমে একটি মান পাওয়া যায়। সাধারণ গণিতশাস্ত্রের মতন এখানেও ঐ প্রক্রিয়াগুলির জন্য একই রকম চিহ্ন ব্যবহার করা হয় ঃ

প্রক্রিয়ার চিহ্ন	প্রক্রিয়ার অর্থ	ব্যবহার	ফলাফল	উদাহরণ
+	যোগ	a + b	a এবং b-র যোগফল	3·5 + 7·2 = 10·7
–	বিয়োগ	a – b	a হইতে b-র বিয়োগফল	8·9 – 6·4 = 2·5 3·2 – 5·6 = –2·4
*	গুণ	a.b	a এবং b র গুফল	1·7 * 3·5 = 5·95
/	ভাগ	a÷b	a কে b দ্বারা ভাগ করলে প্রাপ্ত ভাগফল	3·7 ÷ 1·9 = 7·03
%	মডুলো ভাগ	a % b	a এবং b উভয়েই পূর্ণসংখ্যক চলক এবং প্রাপ্ত ফল হল a-কে b দ্বারা ভাগ করলে প্রাপ্ত ভাগশেষ	7 % 3 = 1

C–ভাষায় গাণিতিক প্রক্রিয়ার ক্ষেত্রে কয়েকটি গুরুত্বপূর্ণ বিষয় উল্লেখ করা দরকার। মনে রাখতে হবে এই প্রক্রিয়া দুরকম ভাবে সম্পন্ন হতে পারে—পূর্ণসাংখ্যিক চলকের সঙ্গে পূর্ণসাংখ্যিক চলকের প্রক্রিয়া এবং বাস্তব চলকের সঙ্গে বাস্তব চলকের প্রক্রিয়া। প্রথম ক্ষেত্রে প্রক্রিয়াগুলিকে আমরা পূর্ণসাংখ্যিক প্রক্রিয়া (Integer Arithmetic) বলতে পারি, দ্বিতীয় ক্ষেত্রে বাস্তব প্রক্রিয়া (Real Arithmetic) বলা যেতে পারে। এছাড়াও বাস্তব চলক ও পূর্ণসাংখ্যিক চলকের উভয়ের মধ্যে প্রক্রিয়াগুলিকে আমরা মিশ্র প্রক্রিয়া বলতে পারি (Mixed Mode Arithmetic)

পূর্ণসাংখ্যিক গাণিতিক প্রক্রিয়া ঃ এই প্রক্রিয়ায় দুইটি পূর্ণসাংখ্যিক চলকের যোগফল, বিয়োগফল, গুণফল, ভাগফল ইত্যাদি নির্ণয় করা হয়। এক্ষেত্রে প্রাপ্ত ফলটিও পূর্ণসাংখ্যিক হয়। ভাগ-প্রক্রিয়ার ক্ষেত্রে অতএব এই বিষয়ে বিশেষ দৃষ্টি দেওয়া প্রয়োজন। উদাহরণ ঃ

মনে করা যাক $a = 16$ এবং $b = 9$ দুইটি পূর্ণসংখ্যটি চলকের মান। অর্থাৎ C–ভাষায় দেওয়া আছে

int a = 16, b = 9;

এখন এদের যোগ/বিয়োগ/গুণের ক্ষেত্রে আমরা সাধারণ গাণিতিক প্রক্রিয়ার মতনই উত্তর পাব—

যেমন ঃ a + b = 25 (পূর্ণসংখ্যা + পূর্ণসংখ্যা = পূর্ণসংখ্যা)

a – 6 = 7 (পূর্ণসংখ্যা – পূর্ণসংখ্যা = পূর্ণসংখ্যা)

অথবা a × b = 144 (পূর্ণসংখ্যা × পূর্ণসংখ্যা = পূর্ণসংখ্যা)

কিন্তু ভাগের ক্ষেত্রে প্রাপ্ত উত্তর হবে ঃ a/b = 16/9 = 1 অর্থাৎ পূর্ণসংখ্যা ÷ পূর্ণসংখ্যা = পূর্ণসংখ্যা এবং ভাগফলের যে ভগ্নাংশ তা কর্তিত (truncated, rounded off নয়) হবে, শুধু মাত্র পূর্ণসাংখ্যিক অংশটি উত্তর হিসাবে পাওয়া যাবে। অতএব দুইটি পূর্ণসংখ্যার মধ্যে ভাগ প্রক্রিয়ার ক্ষেত্রে এই বিশেষ বিষয়টি মনে রাখা দরকন্দার। আরও কয়েকটি উদাহরণ দেওয়া যাক্ ঃ

(i) 8/11 = 0

(ii) (–8) / (–11) = 0

(iii) –8/11 = 0 (কোন কোন পরিগণকে –1 হতে পারে)

Modulo ভাগ ঃ a%b = 16%9 = 7 (ভাগফল = 1, ভাগশেষ = 7)

–18%4 = –2 (প্রাপ্ত ফলের চিহ্ন সর্বদাই প্রথম সংখ্যাটি অর্থাৎ a-র চিহ্নে অনুরূপ হবে)

–18% – 4 = –2

18% – 4 = 2

বাস্তব গাণিতিক প্রক্রিয়া ঃ এই প্রক্রিয়ায় দুইটি বাস্তব চলকের মধ্যে যোগ, বিয়োগ, গুণ বা ভাগ করা হয়, এবং প্রাপ্ত উত্তরটিও বাস্তব মান সম্পন্ন হবে। উদাহরণ ঃ মনে করা যাক দেওয়া আছে

float a = 6·0, b = 8·7, c = –13·1, d = 8·0;

অতএব a/b = 6·0/8·7 = 0·68965517 (rounded of 8sf) (আট সার্থক অঙ্ক পর্যন্ত নির্ভুল—এটাই স্বাভাবিক নির্ভুলতার মাত্রা)

আবার a/b = 6·0/8·0 = 0·75

a/c = 6·0/(–13·1) = –0·45801527

মিশ্র গাণিতিক প্রক্রিয়া ঃ এখানে উভয় প্রকার চলক, পূর্ণসাংখ্যিক বা বাস্তব, ব্যবহৃত হয়। এক্ষেত্রে পূর্ণসাংখ্যিক চলকের মানের শেষে একটি দশমিক বিন্দু বসিয়ে প্রথমে তাকে বাস্তব সংখ্যায় পরিবর্তিত করা হয় (যেমন 13 = 13. বা 13·0, অথবা –11 = –11·0 ইত্যাদি) এবং প্রক্রিয়াটি বাস্তব পদ্ধতিতে সম্পন্ন হয়। যেমন

a = 14·35/2 = 14·35/2·0 = 7·175

কোন রাশিমালার মান নির্ণয়ের জন্য স্বাভাবিক BODMAS–ক্রম ব্যবহার করা হয়। (% এই প্রক্রিয়াটিকে ভাগের সমক্রম হিসাবে ধরা হয়)

(২) সম্পর্কিত প্রক্রিয়া (বা তুলনামূলক প্রক্রিয়া) ঃ

C–ভাষায় সাধারণত ৬ প্রকার তুলনামূলক প্রক্রিয়ার ব্যবহার আছে। সেগুলি হলো ঃ

প্রক্রিয়ার চিহ্ন	অর্থ	ব্যবহার	উদাহরণ দেওয়া আছে float $a = 12{\cdot}4,\ b = 9{\cdot}6;$	চূড়ান্ত ফল
<	ক্ষুদ্রতর	$a < b$	$12{\cdot}4 < 9{\cdot}6$	(সত্য নয়) F বা 0
< =	ক্ষুদ্রতর বা	$a =< b$ অথবা	$12{\cdot}4 <= 9{\cdot}6$	অসত্য অর্থাৎ
অথবা = <	সমান	$a <= b$		False (F) = 0
>	বৃহত্তর	$a > b$	$12{\cdot}4 > 9{\cdot}6$	সত্য অর্থাৎ True (T) = 1
= > অথবা	বৃহত্তর বা সমান	$a => b,$	$12{\cdot}4 \geq 9{\cdot}6$	সত্য/= 1
> =		$a >= b$		
= =	সমান	$a = = b$	$12{\cdot}4 = = 9{\cdot}6$	সত্য নয়/= 0
! =	সমান নয়	$a != b$	$12{\cdot}4 \neq 9{\cdot}6$	সত্য/= 1

এই ধরনের তুলনামূলক প্রক্রিয়ায় ফলস্বরূপ যে উত্তরটি পাওয়া যাবে তা হল সত্য (অর্থাৎ True/সংক্ষেপে T) অথবা সত্য নয় (অর্থাৎ Fale/সংক্ষেপে F)। 'সত্য' অথবা 'সত্য নয়' এই দুই ক্ষেত্রে আউটপুটের সাংখ্যমান যথাক্রমে 1 এবং 0 ধরা হয়ে থাকে। তুলনামূলক প্রক্রিয়ার আরো কিছু উদাহরণ দেওয়া হল যার সাধারণ রূপটি হলে

বীজগাণিতিক রাশিমালা ১ তুলনামূলক প্রক্রিয়া বীজগাণিতিক রাশিমালা ২

যেমন (i) $x + y * z <= x^2 + y^2 + xz$

(ii) sum $<= (x + y)$

(iii) $x + y = = a + b$ (m একটি বর্ণচলক)

(iv) m ! = 'A'

ধরা যাক্, $x = 2{\cdot}5,\ y = 8{\cdot}7,\ z = -3,\ a = 4{\cdot}1,\ b = -1{\cdot}2$

অর্থাৎ int $z = -3$;

float $x = 2{\cdot}5,\ y = 8{\cdot}7,\ a = 4{\cdot}1,\ b = -1{\cdot}2$;

(i) নং উদাহরণে বামদিকের রাশিমালা অর্থাৎ বীজগাণিতিক রাশিমালা ১-এর মান হলো

$$2{\cdot}5 + 8{\cdot}7 \times (-3) = 2{\cdot}5 - 8{\cdot}7 \times 3{\cdot}0$$

$$= 2{\cdot}5 - 26{\cdot}1 = -23{\cdot}6$$

ডানদিকের রাশিমালা $= (2\cdot5)^2 + (8\cdot7)^2 + (2\cdot5)\cdot(-3)$

$= 6\cdot25 + 75\cdot69 - (2\cdot5)\cdot(3\cdot0)$

$= 6\cdot25 + 75\cdot69 - 7\cdot5 = 74\cdot44$

অর্থাৎ প্রদত্ত তুলনামূলক প্রক্রিয়াটিতে লব্দ ফল হল (= 1) সত্য কারণ $-23\cdot6 < 74\cdot44$।

(iii) নং উদাহরণটিতে বামদিকের রাশিমালা $= x + y = 2\cdot5 + 8\cdot7 = 11\cdot2$ এবং = = চিহ্নের ডানদিকের রাশিমালার মান $= a + b$

$= 4\cdot1 - 1\cdot2 = 2\cdot9$

অর্থাৎ $(x + y)$-এর মান এবং $(a + b)$-র তুলনা করে দেখা গেল যে তারা সমান নয়। অতএব এই প্রক্রিয়াটির আউটপুট = 0 (কারণ এটি সত্য নয়)

(৩) যৌক্তিক প্রক্রিয়া (logical operation) ঃ

যৌক্তিক প্রক্রিয়া (logical operation) ঃ C–ভাষাতে তিন প্রকার যৌক্তিক প্রক্রিয়া ব্যবহৃত হতে পারে। এগুলি হলো NOT, AND এবং OR—যৌক্তিক প্রক্রিয়া।

NOT প্রক্রিয়াটি একটিমাত্র তুলনামূলক রাশিমালার উপর কাজ করে। ধরা যাক্ ঐ রাশিমালাটি a দ্বারা চিহ্নিত হল, যেমন $a = x + y * z > x^2 + y^2 + z^2$, এখন x, y এবং z-মান প্রদত্ত হলে a রাশিমালাটির 'মান' অর্থাৎ a সত্য (= 1) না অসত্য (= 0) জানা যাবে। NOT a হলো a-র পূরক (Complement) প্রক্রিয়া, অর্থাৎ a-র 'মান' সত্য হলে NOT a-র মান অসত্য এবং a-র মান অসত্য হলে NOT a-র 'মান' সত্য হবে। C–ভাষাতে NOT প্রক্রিয়ার জন্য ! চিহ্নটি ব্যবহার করা হয়। উদাহরণ ঃ

দেওয়া আছে float $x = 13\cdot2, y = 1\cdot9, z = -6\cdot0$;

অতএব $a = x + y * z > x^2 + y^2 + z^2$

a-র 'মান' অর্থাৎ a সত্য না অসত্য নির্ণয় করতে হবে।

এখন, $x + y * z = 13\cdot2 + 1\cdot9 \times (-6\cdot0)$

$= 1\cdot8$

এবং $x^2 + y^2 + z^2 = (13\cdot2)^2 + (1\cdot9)^2 + (-6\cdot0)^2$

$= 213\cdot85$

অর্থাৎ a মান অসত্য—আউটপুটের মান = 0

অতএব NOT $a = !\ a = !\ \left(x + y * z > x^2 + y^2 + z^2\right) = 0$

AND এবং OR যৌক্তিক প্রক্রিয়া ঃ এই প্রক্রিয়া দুটি যুগ্ম প্রক্রিয়া অর্থাৎ দুটি তুলনামূলক রাশিমালার উপরে কাজ করে। ধরা যাক্ a এবং b দুটি তুলনামূলক রাশিমালা। C–ভাষাতে AND এবং OR—প্রক্রিয়ার চিহ্নগুলি হলো যথাক্রমে a && b এবং a || b

a-র যৌক্তিক মান	b-র যৌক্তিক মান	a AND b (a && b)-র যৌক্তিক মান ও সাংখ্য মান	a OR b (a || b)-র যৌক্তিক ও সাংখ্য মান
F	F	F (সাংখ্যমান = 0)	F (সাংখ্যমান = 0)
F	T	F („)	T (সাংখ্যমান = 1)
T	F	F („)	T („)
T	T	T (সাংখ্যমান = 1)	T („)

উদাহরণ ঃ

(১) দেওয়া আছে

float $x = 67 \cdot 9,\ y = 42 \cdot 3$;

char z = 'D'

মনে করা যাক প্রথম তুলনামূলক রাশিমালাটি হলো $a = (x + y) < z$ এবং দ্বিতীয় তুলনামূলক রাশিমালাটি হলো $b = (x * y / 32 \cdot 4! = z)$ এখন যৌক্তিক রাশিমলা a && b -র যৌক্তিক 'মান' বার করতে হবে—অর্থাৎ এটি সত্য না কি অসত্য অর্থাৎ T না F (সাংখ্যমানের হিসাবে 1 না 0)। এখানে z এই চলকটি একটি বর্ণচলক যার মান D বর্ণটির ASC II কোডের সমতুল্য দ্বিনিধানী সংখ্যাটি অর্থাৎ $(01000100)_2 = (68)_{10}$। এখন যেহেতু $x + y = 110 \cdot 2$ এবং $z = 68$, অতএব a-র যৌক্তিক মান F। আবার $x * y / 32 \cdot 4 = 38 \cdot 6472 \neq 68$ অতএব b-র যৌক্তিক মান T। সুতরাং প্রদত্ত যৌক্তিক রাশিমালাটি হলো a && b এবং এর যৌক্তিকমান হলো F && T = F (অর্থাৎ সাংখ্যিক মান = 0)

(২) এবার যৌক্তিক রাশিমালা a || b -এর যৌক্তিক (এবং সাংখ্যিক) মান নির্ণয় করা যাক।

এক্ষেত্রে a || b -র যৌক্তিক মান F || T = T (অর্থাৎ সাংখ্যিক মান = 1)

(৩) **মিশ্র যৌক্তিক রাশিমালা এবং তার অনুসারক্রম (hierarchy)** এই ধরনের যৌক্তিক রাশিমালাগুলি NOT, AND এবং OR সমন্বয়ে গঠিত হয়, যেমন

$!(x <= y)\ \&\&\ (x + z < y)\ ||\ (x^2 + y^2 == z)$

এক্ষেত্রে প্রথমেই ! (অর্থাৎ NOT)-এর কাজ, পরে && (অর্থাৎ AND)-এর কাজ এবং সবশেষে || (অর্থাৎ OR)—কাজ করতে হবে।

এখানে $(x <= y)$ -এর যৌক্তিক মান F

অতএব $!(x <= y)$-এর যৌক্তিক মান T (প্রথমে NOT-এর কাজ)

$(x + z < y)$-এর যৌক্তিক মান F

$(x^2 + y^2 == z)$-এর যৌক্তিক মান F

অতএব প্রদত্ত যৌক্তিক রাশিমালাটির যৌক্তিক মান হলো

T && F || F (এরপর AND-এর কাজ)

= F || F (শেষে OR-এর কাজ)

= F

(৪) মান প্রদানকারী প্রক্রিয়া

এই প্রক্রিয়ার সাহায্যে কোন চলকের মান কোন ধ্রুবক বা কোন রাশিমালার মানের সঙ্গে সমান করা হয়। চলকটির পূর্বতন কোন মান (যদি থাকে) তবে তা এই নতুন মানের দ্বারা প্রতিস্থাপিত হবে, পূর্বতন মানটির কোন অস্তিত্ব থাকবে না। এই কাজগুলি করা যেতে পারে নীচের নির্দেশগুলির সাহায্যে ঃ

(i) $x = 4 \cdot 9$; [চলক x-এর মান 4·9 করা হল।

(ii) $x = y + z * z$; [y এবং z-এর প্রদত্ত মানের সাহায্যে প্রথমে $y + z * z$-এর মান নির্ণিত হবে এবং সেই মানটি x-এর মান হিসাবে পরিগণিত হবে]

(iii) $x = y = z = 12 \cdot 4$;

$a = b = c = 1 \cdot 2 + x + 7 \cdot 2 * y - z * 3$

[এর ফলে $a = b = c = 65 \cdot 68$ হবে/যাচাই করে দেখুন]

(iv) সংক্ষিপ্ত আকারে + =, – = ইত্যাদি মানপ্রদানকারী প্রক্রিয়া ঃ

প্রক্রিয়া (সংক্ষিপ্ত আকারে)	ব্যবহার	অর্থ
+ =	$x+ = 4 \cdot 2$	$x = x + 4 \cdot 2$
– =	$x- = 4 \cdot 2$	$x = x - 4 \cdot 2$
* =	$x* = 6$	$x = x * 6$
/ =	$a/ = 4 \cdot 0$	$a = a / 4 \cdot 0$
% =	$x\% = (y + z)$	$x = x\% \ (y + z)$

বিঃ দ্রঃ- সাধারণ বীজগণিতে $x = x + 4 \cdot 2$ অশুদ্ধ একটি সমীকরণ, কিন্তু পরিগণকের ভাষায় এর অর্থ হলো

x-চলকের বর্তমান মানের সঙ্গে 4·2 যোগ করে যে মান পাওয়া গেল সেটিই এখন থেকে x-এর বর্তমান মান হিসাবে গণ্য হবে, পূর্বের মানটির অস্তিত্ব থাকবে না।

(৫) বৃদ্ধিকরণ এবং হ্রাসকরণ প্রক্রিয়া

এই প্রক্রিয়া দুটি ++ এবং −− এই চিহ্নদ্বারা প্রকাশ হয়। এর অর্থ হলো ঃ

n ++ (অথবা ++ n) : $n = n + 1$

n −− (অথবা −− n) : $n = n - 1$

অর্থাৎ এই প্রক্রিয়ার সাহায্যে কোন চলকের বর্তমান মানকে 1 বাড়ানো বা কমানো যেতে পারে।

(৬) শর্তাধীন প্রক্রিয়া

C–ভাষায় এটি একমাত্র এমন একটি প্রক্রিয়া যা তিনটি রাশিমালার মধ্যে ক্রিয়াশীল। রাশিমালা তিনটি যদি expr1, expr2, এবং expr3 দ্বারা সূচিত করা হয় তবে এই প্রক্রিয়াটির সাধারণ রূপ হলো

expr1? expr2 : expr3 ;

এই প্রক্রিয়ার প্রথমেই expr1-এর যৌক্তিক মান নির্ণয় করা হয়। যদি সেটি TRUE হয় তবে expr2-র নির্দেশ কার্যকরী হবে, অন্যথায় (অর্থাৎ expr1-রে যৌক্তিক মান FALSE হলে) expr3-এর নির্দেশ কার্যকরী হবে।

উদাহরণ (১) ঃ $(n <= 100)?\ n = n + 1 :$ sum = sum + n ;

এখানে expr1 হলো $n <= 100$

expr2 হলো $n = n + 1$ এবং expr3 হলো sum = sum + n

n-এর মান ≤ 100 হলে, $n = n + 1$ হবে, অন্যথায় $n > 100$ হলে sum = sum + n হবে।

উদাহরণ (২) ঃ int $x = 12,\ y = 14$;

$$a = (x <= y)\ ?\ x : y$$

এখানে expr1 হলো $x <= y$ এবং expr2, expr3 হলো যথাক্রমে $a = x$ এবং $a = y$

প্রদত্ত মান অনুযায়ী $x < y$ অর্থাৎ expr1 = T, অতএব $a = x = 12$ হবে। পরবর্তী পর্যায়ে আমরা দেখব এই প্রক্রিয়াটি নীচের প্রোগ্রাম অংশের সমতূল্য ঃ

if $(x <= y)$ then

$a = x$;

else

$a = y$;

C–ভাষার আপেক্ষক ভাণ্ডার (Library Functions)

অন্যান্য পরিগণকীয় ভাষার ন্যায় C–ভাষাও এক বিশাল অপেক্ষক ভাণ্ডারে সমৃদ্ধ। সহজে নির্দেশিকা তৈরী করার জন্য ব্যবহারকারীর সুবিধার্থে বহুল ব্যবহৃত বেশ কিছু অপেক্ষক নিয়েই এই ভাণ্ডার। এই আপেক্ষকগুলির একটা বড় অংশ গাণিতিক/ত্রিকোনমিতির অপেক্ষক, কিছু অপেক্ষক ইন্‌পুট বা আউটপুটের কাজে ব্যবহার করা হয়। অন্যান্য কিছু অপেক্ষক আছে যাদের বিভিন্ন তন্ত্রীধ্রুবক/তন্ত্রীচলকের মধ্যে সমন্বয় সাধনের কাজে ব্যবহার করা হয়। তাছাড়া বর্ণধ্রুবক/চলক সংক্রান্ত কিছু অপেক্ষকও আছে এই ভাণ্ডারে। C–ভাষার এই অপেক্ষক ভাণ্ডার এতটাই দীর্ঘ যে তার সম্পূর্ণ তালিকা প্রকাশ করতে বেশ কিছু পৃষ্ঠার প্রয়োজন হতে পারে। এখানে ঐ ভাণ্ডারের বহুল ব্যবহৃত কিছু অংশ সন্নিবিষ্ট করা হলো ঃ

C–অপেক্ষক $f(x)$	অর্থ	স্বাধীন চলরাশি (x)-এর প্রকৃতি	অপেক্ষক $f(x)$-এর মানের প্রকৃতি
sqrt (x)	$\sqrt{x}$	বাস্তব সংখ্যা (≥ 0)	(≥ 0) বাস্তব সংখ্যা
$\log(x)$	$\log_e(x)$	বাস্তব সংখ্যা (>0)	বাস্তব সংখ্যা
$\log 10(x)$	$\log_{10}(x)$	বাস্তব সংখ্যা (>0)	বাস্তব সংখ্যা
abs (x)	$\|x\|$	শুধুমাত্র পূর্ণসংখ্যা	পূর্ণসংখ্যা (≥ 0)
fabs (x)	$\|x\|$	যে কোন বাস্তব সংখ্যা	বাস্তব সংখ্যা (≥ 0)
exp (x)	e^x	বাস্তব সংখ্যা	বাস্তব সংখ্যা
pow (x,y)	x^y	x, y উভয়েই বাস্তব (x ঋনাত্মক হলে y ভগ্নাংশ হবে না)	বাস্তব
pow $10(x)$	10^x	x পূর্ণসংখ্যা	পূর্ণসংখ্যা
ceil(x)	x-এর মানের ঠিক পরবর্তী পূর্ণসংখ্যা অর্থাৎ ক্ষুদ্রতম পূর্ণসংখ্যা $\geq x$	বাস্তব (মূলদ)	শুধুমাত্র পূর্ণসংখ্যা উদাহরণ ঃ ceil (–8·15) = – 8 ceil (6·03) = 7
fmod (x,y)	$(x \div y)$-এর অবশিষ্ট অর্থাৎ ভাগফলের ভগ্নাংশ (x-এর চিহ্নযুক্ত)	x, y বাস্তব	বাস্তব (মূলদ) উদাহরণ ঃ fmod (–4·89, 1·28) = – 0·8203125

ত্রিকোণমিতিক অপেক্ষক			
$\sin(x)$	$\sin x$	বাস্তব (রেডিয়ানে)	বাস্তব
$\cos(x)$	$\cos x$	বাস্তব (রেডিয়ানে)	বাস্তব
$\tan(x)$	$\tan x$	বাস্তব (রেডিয়ানে)	বাস্তব
$a\sin(x)$	$\sin^{-1}(x)$	x বাস্তব এবং $-1 \le x \le 1$	বাস্তব (রেডিয়ানে) $(-\pi/2 \le a\sin(x) \le \pi/2$
$a\cos(x)$	$\cos^{-1}(x)$	x বাস্তব এবং $-1 \le x \le 1$	বাস্তব (রেডিয়ানে) $0 \le a\cos x \le \pi/2$
$a\tan(x)$	$\tan^{-1}(x)$	x বাস্তব	বাস্তব (রেডিয়ানে) $-\pi/2 \le a\tan x \le \pi/2$
$a\tan 2(x,y)$	$\tan^{-1}(x/y)$	x, y বাস্তব	বাস্তব (রেডিয়ানে)
পরাবৃত্তীয় অপেক্ষক			
$\cos h(x)$	$\cos h\, x$	বাস্তব	বাস্তব
$\sin h(x)$	$\sin h\, x$	বাস্তব	বাস্তব
$\tan h(x)$	$\tan h\, x$	বাস্তব	বাস্তব
বর্ণগত অপেক্ষক			
toascii (x)	বর্ণচলক x-এর সাংখ্যিক মান	বর্ণচলক	পূর্ণসংখ্যা
tolower (x)	বড় হাতের বর্ণকে ছোট হাতে রূপান্তর	বর্ণচলক	বর্ণচলক
toupper (x)	ছোট হাতের বর্ণকে বড় হাতে রূপন্তর	বর্ণচলক	বর্ণচলক

উপরের এই তালিকাটি C–অপেক্ষক ভাণ্ডারের একটি অংশ মাত্র। নীচের কয়েকটি উদাহরণের মাধ্যমে এদের ব্যবহার আরো পরিস্কার করে দেখানো হলো ঃ

(i) abs : abs (– 12) = 12
abs (2) = 2

abs $(x+y) = |x+y|$ (x, y পূর্ণসংখ্যা)

abs $(8+4) = |12| = 12$

abs $(5-14) = |-9| = 9$

(ii) fabs : fabs $(-12\cdot5) = 12\cdot5$

fabs $(18\cdot2+17\cdot9) = |18\cdot2+17\cdot9| = 36\cdot1$

(iii) pow : pow $(3\cdot5,7\cdot3) = (3\cdot5)^{7\cdot3}$ $2^3 = \text{pow}(2\cdot0,3\cdot0)$ [pow (2, 3) হবে না]

pow $(-5\cdot0, 1\cdot5)$ গ্রাহ্য নয়।

মনে রাখতে হবে pow (x, y) -তে x এবং y উভয়েই বাস্তব (অর্থাৎ floating) হতে হবে। এই কারণে pow (3, 2) বৈধ নয়।

pow $(x+y, 2\cdot0) = (x+y)^2$

(iv) ceil : ceil $(8\cdot6) = 9$

ceil $(-4\cdot0) = -4$

ceil $(9\cdot0) = 9$

ceil $(-8\cdot6) = -8$

ceil $(-0\cdot8) = 0$

দ্রষ্টব্য ceil (x)-এ x-এর মান বাস্তব (floating) প্রকৃতির হতে হবে

(v) fmod : fmod $(8\cdot0, 2\cdot0) = 0\cdot0$

fmod $(8\cdot0, 3\cdot0) = 2\cdot0$

fmod $(3\cdot0, 8\cdot0) = 3\cdot0$

(vi) toascii : toascii $(\text{A}) = 65$

toascii $(8) = 56$

toascii $(d) = 100$

toascii $(+) = 43$

(vii) tolower : tolower $('\text{A}') = a$

tolower $(68) = \text{tolower}('\text{D}') = d$

C–ভাষাতে ইন্‌পুট এবং আউটপুট সংক্রান্ত বেশ কিছু অপেক্ষক আছে যেগুলি প্রায়শই ব্যবহৃত হয়ে থাকে। এদের মধ্যে বিশেষ কয়েকটি হলো ঃ

ইন্‌পুট সংক্রান্ত : gets : তন্ত্রীচলক বা ধ্রুবক পড়বার জন্য ব্যবহৃত হয়
getchar : একক বর্ণ ধ্রুবক/চলকের মান পড়বার জন্য
scanf : যে কোন ধরনের সাংখ্যিক চলক বা ধ্রুবক বা এক বর্ণ ধ্রুবক/চলকের মান পড়বার জন্য।

আউটপুট সংক্রান্ত অপেক্ষক ঃ
puts : আউটুট হিসাবে প্রাপ্ত বর্ণচলক ধ্রুবকের মান মুদ্রণের জন্য
putchar : একক বর্ণ চলকের মুদ্রণ
printf : যে কোন চলক বা ধ্রুবকের মান মুদ্রণের জন্য

হেডার ফাইল (Header File)

C–ভাষার Header File-কে আমরা সমস্ত অপেক্ষক ভাণ্ডারের মস্তিষ্ক হিসাবে গণ্য করতে পারি, যেখানে ভাণ্ডারের অপেক্ষকগুলি পরপর ফাইল হিসাবে সজ্জিত থাকে। মস্তিস্কে এই ফাইলগুলি তাদের প্রকৃতি অনুসারে বিভিন্ন শ্রেণীতে ভাগ করে রাখা হয়। যেমন, গাণিতিক ফাইলগুলি, যেমন $\sin(x)$, $\cos(x)$, $\exp(x)$, $\text{sqrt}(x)$ ইত্যাদি, গাণিতিক হেডার ফাইলে সন্নিবিষ্ট আছে যার নাম math·h (অর্থাৎ mathematical header file · h হেডার ফাইল সূচিত করে)। আবার ইনপুট-আউটপুট সংক্রান্ত ফাইলগুলি, যেমন putchar, getchar, scanf, printf ইত্যাদি, প্রামান্য ইনপুট-আউটপুট হেডার ফাইলে রাখা আছে যার সংক্ষেপিত নাম হলো stdio·h (অর্থাৎ standard input-output header file)। নীচের সারণীতে বিভিন্ন ধরণের হেডার ফাইলের একটি সংক্ষিপ্ত বর্ণনা দেওয়া হলো ঃ

হেডার ফাইলের নাম	ঐ ফাইলে অন্তর্ভূক্ত কিছু অপেক্ষক	
stdio·h	gets puts getchar putchar scanf printf	
math·h	sin cos tan log log 10 exp pow pow 10	asin acos a tan a tan2 sinh cosh tanh ceil fmod fabs

string·h	strcat (দুই বা অধিক তন্ত্রকে যুক্ত করার জন্য)
(তন্ত্রী সংক্রান্ত	strcmp (দুটি তন্ত্রকে অভিধানিক ক্রম অনুসারে তুলনা করার জন্য)
হেডার ফাইল)	strcpy (তন্ত্রীচলকের অনুলিপিকরণের জন্য)
	strlen (তন্ত্রীচলকের দৈর্ঘ্য অর্থাৎ বর্ণসংখ্যা নির্ণয়ের জন্য)
ctype·h	toascii
(বর্ণচলক সংক্রান্ত	tolower
হেডার ফাইল)	toupper

এই তালিকাটি সম্পূর্ণ নয়, এছাড়াও আরো অনেক হেডার ফাইল আছে যার মধ্যে অন্যতম হলো stdlib·h (যার মধ্যে অন্যতম হলো abs (i), rand ( ), atoi (s)—যা কোন তন্ত্রীকে পূর্ণসংখ্যায় পরিবর্ত্তিত করে, ইত্যাদি), time·h (সময় নির্ধারনের জন্য), graphics · h (বিভিন্ন নক্সা ইত্যাদি তৈরী করার জন্য)।

C–নির্দেশিকায় (C–program) এই অপেক্ষক ভাণ্ডারের ব্যবহার অপরিহার্য। কোন নির্দেশিকায় যে অপেক্ষক (বা অপেক্ষক সমূহ) টি ব্যবহার করা হবে, সেই অপেক্ষকটি যে হেডার ফাইলে আছে, প্রথমেই সেই হেডার ফাইলটিকে নির্দেশিকার অন্তর্ভূক্ত করা প্রয়োজন। উদাহরণস্বরূপ মনে করা যাক্ কোন C–নির্দেশিকায় scanf ( ) বা printf ( ) অপেক্ষকগুলির ব্যবহৃত হয়েছে, সেক্ষেত্রে এই অপেক্ষকগুলি যে হেডার ফাইলে আছে, অর্থাৎ stdio · h তাকে নির্দেশিকার প্রথমেই অন্তর্ভূক্ত করে নিতে হবে। আবার C–নির্দেশিকার সাহায্যে কোন গাণিতিক সমস্যার সমাধানে ধরা যাক $\sin(x)$, $\cos(x)$, $\exp(x)$ $\text{sqrt}(x)$ বা এই ধরণের কোন গাণিতিক অপেক্ষক ব্যবহার করা হয়েছে, সেক্ষেত্রে নির্দেশিকায় math · h-এর অন্তর্ভূক্তি অবশ্যই প্রয়োজন। হেডার ফাইলের এই অন্তর্ভূক্তিকরণের জন্য যে নির্দেশটির প্রয়োজন, তা হলো ঃ

```
# include < stdio · h >
```

অথবা

```
# include < math · h >
```

এই অন্তর্ভূক্তিকরণ নির্দেশিকার প্রথমেই করতে হবে। # চিহ্নটি প্রথম কলমেই লিখতে হবে।

C–ভাষায় পাটীগণিতিক বিবৃত-করণ (Arithmetic Statement in C) ঃ

অন্যান্য উচ্চস্তরের পরিগণকীয় ভাষার মতন C–ভাষাতেও পাটিগাণিতিক বিবৃত-করণের সাধারণ রূপটি হলো ঃ

চলক = রাশিমালা

যেমন $x = a + b * c$;

$s = s + n$

$m = x / y + \text{pow}\ (a, b)$

$d = \text{sqrt}\ \left(\text{abs}(d * d - 4 * b * c)\right)$ ইত্যাদি,

নীচের উদাহরণগুলি বৈধ C–বিবৃতি নয় ঃ

$x + y = a + b$ (বাঁদিকে শুধুমাত্র একটি C–চলক থাকবে)

$4 \cdot 25 = x + y * z$ (বাঁদিকে চলক থাকবে, ধ্রুবক নয়)

এ প্রসঙ্গে কয়েকটি মন্তব্য বিশেষভাবে প্রণিধান যোগ্য : বৈধ C–বিবৃতিতে ব্যবহৃত সাংখ্যিক চলকগুলির প্রকৃতি অনুসারে বিভিন্ন সম্ভাবনার উদ্ভব হতে পারে—

(১) পূর্ণসাংখ্যিক চলক = পূর্ণসাংখ্যিক রাশিমালা

যেমন, int a,b,c;

$a = b - 3 * c + 50$;

এখন, $b = 4, c = 5$ হলে $a = 4 - 15 + 50 = 39$ হবে, অর্থাৎ ডান দিকের রাশিমালাটির মান পূর্ণসংখ্যার হবে, এবং সেই পূর্ণসংখ্যাটি a-এর মান হিসাবে সঞ্চিত হবে।

(২) পূর্ণসাংখ্যিক চলক = বাস্তব রাশিমালা

যেমন, int a ;

float b,c,d ;

$a = b * c - 3 \cdot 25 * d$

এখন $b = 1 \cdot 25,\ c = -7 \cdot 17$ এবং $d = 1 \cdot 47$ হলে ডানদিকের বাস্তব রাশিমালার মান বাস্তব পদ্ধতিতে (floating-point arithmetic) নির্ণীত হবে এবং ঐ রাশিমালাটির মান হবে = $1 \cdot 25 \times (-7 \cdot 17) - 3 \cdot 25 \times (1 \cdot 47) = -13 \cdot 74$, কিন্তু যেহেতু বাঁদিকে পূর্ণসাংখ্যিক চলক আছে, অতএব a-র মান হবে – 13, ভগ্নাংশ ·74 কর্তিত হবে। অতএব, এই প্রকার বিবৃতির ক্ষেত্রে আমাদের সতর্ক থাকা প্রয়োজন।

(৩) বাস্তব চলক = পূর্ণসাংখ্যিক রাশিালা

উদাহরণ int a,b,c

float d

$d = b * b - 4 * a * c$

ডানদিকের রাশিমালাটির মান পূর্ণসাংখ্যিক হবে, এবং ঐ পূর্ণসংখ্যার ডানদিকে একটি দশমিক বিন্দু বসিয়ে তাকে বাস্তব সংখ্যায় পরিবর্তিত করা হবে, এবং সেই পরিবর্তিত বাস্তব সংখ্যাটিই হবে d-র মান।

ধরা যাক $a = 2, b = 9, c = 3$

ডানদিক $= 9 \times 9 - 4 \times 2 \times 3 = 81 - 24 = 57$ (পূর্ণসংখ্যা)

এবং বামদিক $= d = 57 \cdot 0$ (বাস্তব সংখ্যা)

(৪) বাস্তব চলক = বাস্তব রাশিমালা

উদাহরণ float x, y, z, t ;

$$t = x * y + 3 \cdot 25 * z;$$

এক্ষেত্রে ডানদিকের রাশিমালাটির মান বাস্তব অর্থাৎ floating-point পদ্ধতিতে নির্ণীত হবে, প্রাপ্ত মান হবে বাস্তব এবং তা বাঁদিকের বাস্তব চলকের মান হিসাবে চিহ্নিত হবে।

সংক্ষিপ্তসার

এই পাঠে পরিগণকের বিভিন্ন ধরণের ভাষার সংক্ষিপ্ত বর্ণনা দেওয়া হয়েছে। C–ভাষার সৃষ্টির ইতিহাস ও এখানে লেখা হয়েছে। C–ভাষায় ব্যবহৃত চিহ্ন এবং বিশেষ অর্থবহ শব্দগুলি তালিকা আকারে দেওয়া আছে। C–ভাষায় ব্যবহৃত বিভিন্ন ধ্রুবক এবং চলরাশির বর্ণনা দেওয়া আছে। চলকের নামকরণের নিয়ম ও প্রকৃতি কিভাবে নির্ধারণ করা হয় তা আলোচনা করা হয়েছে, C–ভাষার বিভিন্ন প্রক্রিয়া, যেমন—যোগ, বিয়োগ, গুণ, ভাগ ইত্যাদির নিয়ম এবং বীজগণিত রাশিমালা C–ভাষায় লেখার নিয়মাবলীও এখানে দেওয়া আছে। গণিতে ব্যবহৃত অপেক্ষক গুলিকে C–ভাষার আপেক্ষক ভাণ্ডারে একসঙ্গে রাখা হয়েছে। তাদের ব্যবহার করার নিয়মাবলী এবং ব্যবহার উদাহরণসহ বর্ণনা করা হয়েছে।

একক 6 ❑ C —নির্দেশাবলী

এককের উদ্দেশ্য ঃ

এই একক পাঠ করলে পাঠক নিম্নলিখিত বিষয়গুলি সমন্ধে অবগত হবে—

- C–ভাষায় নির্দেশাবলী লেখার নিয়মাবলী
- ইনপুট ও আউটপুট অপেক্ষক
- বিভিন্ন ধরণের if বিবৃতি
- goto বিবৃতি
- switch প্রক্রিয়া নির্দেশক বিবৃতি
- while–লূপ এবং do-while লূপ
- for–লূপ ও তার ব্যবহার
- array–চলক

C–ভাষায় নির্দেশাবলী লেখার প্রাথমিক কিছু তথ্য ঃ

কিভাবে C–নির্দেশ লিখবেন? এই নির্দেশাবলী লেখার আগে কতকগুলি প্রাথমিক তথ্য জেনে নেওয়া প্রয়োজন।

(১) C–নির্দেশাবলী লেখার জন্য C-র ধ্রুবক, চলক এবং তার বিশেষ শব্দ ও অপেক্ষকগুলিই কেবলমাত্র ব্যবহার করা যাবে।

(২) প্রতিটি নির্দেশ এক একটি লাইনেই লিখতে হবে।

(৩) C–একটি স্বাধীন রূপের ভাষা, FORTRAN-এর মতন বদ্ধ নয়, অর্থাৎ C-র নির্দেশ লেখার কাজ যে কোন Column থেকেই শুরু করা যায়।

(৪) বিশেষ কয়েকটি বিবৃতি ছাড়া (যেগুলি সম্বন্ধে আমরা পরে জানবো) প্রতিটি বিবৃতির শেষ হবে সেমিকোলন (;) চিহ্ন দিয়ে।

(৫) সাধারণভাবে C–নির্দেশাবলী ছোট হাতের অক্ষরে লেখা হয়, তবে ফাইলের নাম ইত্যাদি ক্ষেত্রে বড় হাতের হরফ ব্যবহার করা যেতে পারে।

(৬) হেডার ফাইল অন্তর্ভূক্তির জন্য ব্যবহৃত # চিহ্নটি প্রথম কলমেই লিখতে হবে।

(৭) C–ভাষায় লেখা কোন নির্দেশিকায় প্রয়োজন হলে বিশেষ কোন মন্তব্য অন্তর্ভূক্ত করা যায়। ধরা যাক

Newton-Raphson পদ্ধতিরে সাহায্যে একটি সমীকরণের বীজ নির্ণয়ের জন্য C–ভাষায় একটি নির্দেশাবলী লেখা হলো। এই নির্দেশাবলীর একটি শিরোনাম থাকলে পরবর্তীকালে কিসের জন্য এই নির্দেশাবলীটি লেখা হয়েছিল, শুধুমাত্র শিরোনামের সাহায্যেই তা বোঝা যেত। তাছাড়া দীর্ঘ কোন নির্দেশাবলীর ক্ষেত্রে অন্তবর্তী কোন কোন মন্তব্য নির্দেশাবলীটি যথাযথভাবে অনুধাবন করতে যথেষ্ট সহায়ক হয়। এই ধরণের মন্তব্যগুলি লেখার শুরুতে /* এবং শেষ */ চিহ্নদুটি ব্যবহার করতে হবে।

যেমন /* Program for Newton-Raphson method */ মন্তব্যটি একাধিক লাইনেও বিস্তৃত হতে পারে, যেমন,

```
/* Program written by : S. K. Dey
   Year : B. Sc. 1st year
   Roll : 37 */
```

অথবা

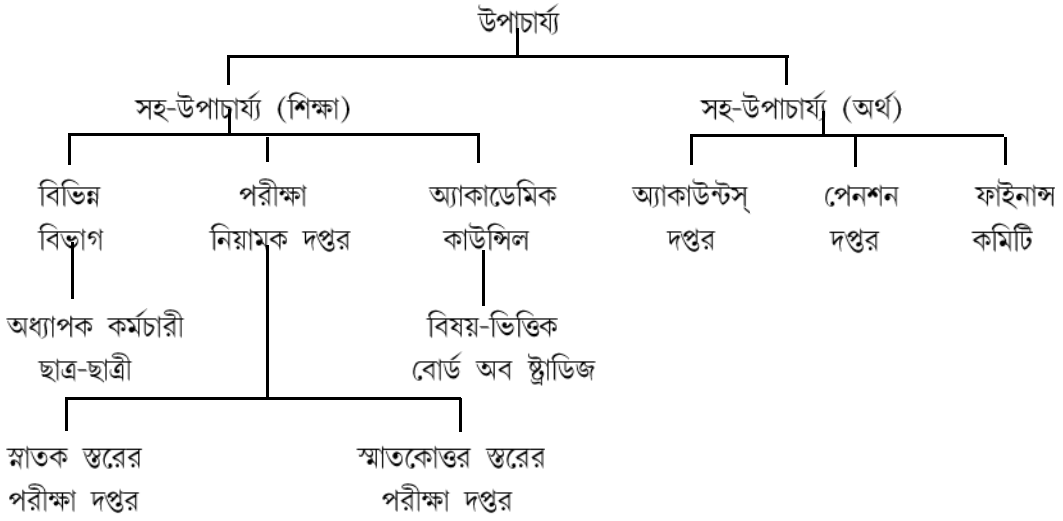
```
/* Program written by : S. K. Dey */
/* Year : B. Sc. 1st year */
/* Roll : 37 */
```

তবে কোন মন্তব্যের অভ্যন্তরে অন্য কোন মন্তব্য করা যাবে না। যেমন /* Program written by : S. K. Dey /* Year : B. Sc. 1st year /* Rill-37 */ —এটি বৈধ নয়।

মন্তব্য-সূচক এই বিবৃতিগুলি শুধুমাত্র পড়বার জন্য এবং যথাযথ স্থানে অবিকৃতভাবে তা লেখার জন্যই ব্যবহৃত হয়, অন্য কোন প্রক্রিয়ায় এগুলি কোনরূপ অংশগ্রহণ করে না (non-executable)

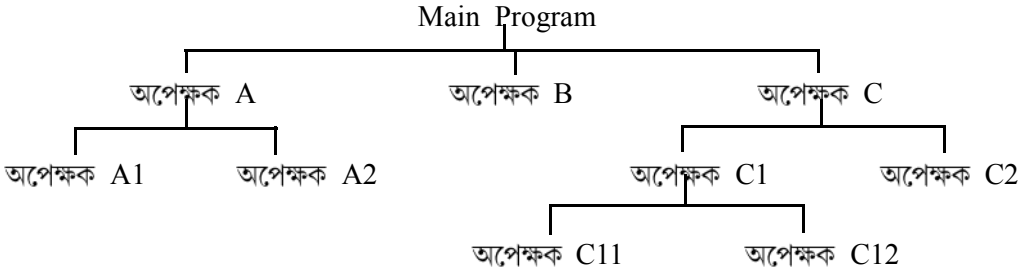
C–ভাষায় নির্দেশিকা লেখার পদ্ধতিটে module–ভিত্তিক, অর্থাৎ সমগ্র নির্দেশিকাটি ছোট ছোট অংশে (অর্থাৎ module-এ) বিভাজিত। এক একটি অংশ এক একটি কাজের জন্য সুনির্দ্দিষ্ট। এই কারণে এই পদ্ধতিটিকে বলা হয় Modular Programming—কোন প্রতিষ্ঠানের সাংগঠনিক পরিকাঠামোর সঙ্গে এর তুলনা করা যেতে পারে।

উদাহরণস্বরূপ কোন বিশ্ববিদ্যালয়ের সাংগঠনিক কাঠামোর কথা ধরা যেতে পারে ঃ



এখানে প্রতিটি বিভাগের প্রধান অধ্যাপক তাঁর সহকর্মী অধ্যাপক ও অশিক্ষক কর্মচারীদের সাহায্যে বিভাগীয় কার্য সম্পন্ন করবেন এবং বৎসরান্তে সহ-উপাচার্য্য (শিক্ষা)-র কাছে তাঁর কাজের রিপোর্ট পেশ করবেন। পরীক্ষা-নিয়ম বিভাগের কাজ হলো সমস্ত পরীক্ষা নেওয়া ও তার ফলাফল প্রকাশ করা। বিষয়গত বোর্ড অব ষ্টাডিজ ঐ বিষয়ের পাঠ্যসূচী/পরীক্ষকদের নাম/প্রশ্নপত্রের ধরণ ইত্যাদি ব্যাপারে সিদ্ধান্ত নেবেন। অ্যাকাডেমিক কাউন্সিল সামগ্রিকভাবে সমস্ত বিষয়ের বোর্ড অব ষ্টাডিজের সিদ্ধান্তগুলি একত্রিত করে সহ-উপাচার্য্য (শিক্ষা)-র কাছে পেশ করবেন। অনুরূপভাবে সহ-উপাচার্য্য (অর্থ), অর্থনৈতিক সংক্রান্ত সমস্ত বিষয়গুলি পরিচালনা করবেন। সামগ্রিক দায়িত্বে থাকবেন উপাচার্য্য, যিনি বিশ্ববিদ্যালয়ের কৃতির মূল দায়িত্বপ্রাপ্ত। অর্থাৎ প্রতিটি অংশ তার নিজ নিজ কাজ সম্পন্ন করবে এবং সেই অংশগত ফলগুলি সবশেষে পৌঁছবে সংস্থার সর্বোচ্চ ব্যাক্তির নিকট—যাঁর কাছ থেকে সংস্থার সামগ্রিক কৃতির এক খতিয়ান পাওয়া যাবে।

C–ভাষার নির্দেশিকার চরিত্রটিও অনেকটা এইরকম। C–ভাষায় এই প্রধান অপেক্ষকটি main ( ) দ্বারা সূচিত হয় (main এই শব্দটির পরে একটি বাম বন্ধনী ও একটি ডান বন্ধনী)। প্রতিটি নির্দেশিকায় একটিই main অপেক্ষক অবশ্যই থাকবে, এবং main-এর অবস্থান নির্দেশিকাটির সূচনা নির্দেশ করে। main অপেক্ষকের অধীনে অন্যান্য অপেক্ষাকও থাকতে পারে। অর্থাৎ নির্দেশিকাটির চিত্ররূপটি হলো ঃ



এই চিত্ররূপটির কার্যপ্রণালী এভাবে ব্যাখ্যা করা যায় ঃ

C11–অপেক্ষকটি তার নির্দেশিকা অনুসারে কাজ করবে এবং একটি আউটপুট তৈরী করবে। অনুরূপভাবে C12-ও একটি আউটপুট তৈরী করবে। এই আউটপুট দুটি জমা পড়বে C1–অপেক্ষকের কাছে। C1 অপেক্ষকটি এই দুটি আউটপুট ব্যবহার করে তার নির্দেশিকা অনুসারে কাজ করবে এবং একটি আউটপুট তৈরী করবে, সেটি জমা পড়বে অপেক্ষক C-র কাছে। আবার অন্যদিকে অপেক্ষক C2-র আউটপুটও জমা পড়বে C-র কাছে আপেক্ষক C, C1 এবং C2-র জমা দেওয়া আউটপুট ব্যবহার করে তার নিজস্ব নির্দ্দিষ্ট কাজটি সম্পন্ন করবে, এবং উৎপন্ন আউটপুটটি main-এর কাছে পেশ করবে। main-এর অন্যান্য অপেক্ষক অংশগুলিও (অর্থাৎ অপেক্ষক A এবং B) অনুরূপভাবে তাদের আউটপুট main-এর কাছে পেশ করবে। main অপেক্ষকটি এবার এই সমস্ত পেশ করা আউটপুটগুলির সাহায্যে নির্দেশিকার মূল নির্দেশগুলি অনুসরণ করে অন্তিম বা final আউটপুট তৈরী করবে।

C–তে ব্যবহৃত অপেক্ষকগুলি দু ধরণের হতে পারে—পূর্বে উল্লেকিত অপেক্ষক-ভাণ্ডারের অন্তর্গত কোন অপেক্ষক অথবা ব্যবহারকারী কর্তৃক নিজস্ব প্রয়োজনে সৃষ্ট কোন অপেক্ষক (user-defined function)। main অপেক্ষকটি এই ধরণেরই একটি অপেক্ষক, কিন্তু printf, scanf এই অপেক্ষকগুলি হেডার ফাইলের অন্তর্গত অপেক্ষক (Library Function)। প্রকৃতপক্ষে C–নির্দেশিকা হলো কয়েকটি অপেক্ষকরে সমন্বয়, যার একটি অপেক্ষক

অবশ্যই main। কোন নির্দেশিকাতে কেবলমাত্র একটিই main থাকবে, অন্য অপেক্ষকও থাকতে পারে। এরকম নির্দেশিকার সাধারণরূপ হলো ঃ

```
main ( )
{              /* অপেক্ষক main-এর সূচনা */
statement 1;
statement 2;
}              /* অপেক্ষক main-এর শেষ */
```

অর্থাৎ ঐ অপেক্ষকের অন্তর্ভুক্ত সমস্ত বিবৃতিই একজোড়া ধনুর্বন্ধনীর মধ্যে আবদ্ধ থাকবে।

ইন্‌পুট এবং আউটপুট অপেক্ষক (Input and Output function)

ইন্‌পুট এবং আউটপুট এর জন্য C–ভাষাতে যে অপেক্ষক দুটি সবচেয়ে বেশী ব্যবহৃত হয়, সেগুলি হলো scanf এবং printf—এই দুটিই stdio · h—এই হেডার ফাইলের অন্তর্গত।

ইনপুট অপেক্ষক scanf-এর সাধারণ রূপ ঃ

scanf (''চলকগুলির প্রকৃতি নির্দেশক তন্ত্রী'', প্রথম চলকের Address, দ্বিতীয় চলকের Address, ...);

এখানে চলকগুলির প্রকৃতি বিভিন্ন প্রকার হতে পারে, যেমন দশমিক পূর্ণসংখ্যা, দশমিক বাস্তব (অর্থাৎ floating point) এককমাত্রার বর্ণচলক, তন্ত্রীচলক প্রভৃতি। এই প্রকৃতিগুলি বোঝানোর জন্য যে চিহ্নগুলো ব্যবহার করা হয় সেগুলি হলো

% c : একক বর্ণ চলকের প্রকৃতি নির্দেশক

% f বা % e বা % g : (floating-point) বাস্তব চলক নির্দেশক

% d : দশমিক পূর্ণসাংখ্যিক চলকের জন্য

% s : তন্ত্রী চলক নির্দেশক

চলকগুলির Address সম্বন্ধে একেবারে প্রথমেই আলোচনা করা হয়েছে (একক 1..... দ্রষ্টব্য)। কোন চলক *x*-এর Address নির্দেশ করার জন্য & *x* অর্থাৎ চলকটির পূর্বে & চিহ্নটি বসাতে হবে। (তন্ত্রীচলকের ক্ষেত্রে &-চিহ্নটির প্রয়োজন নেই)

ধরা যাক্, *x*, *y* এবং *z* তিনটি চলক এবং এদের প্রকৃতি হলো ঃ

```
int x ;
float y ;
char z ;
```

scanf—অপেক্ষকের মাধ্যমে এই চলকগুলির মান পরিগণককে জানাতে হলে, আমরা লিখবো ঃ

scanf (" % d % f % c", & *x*, & *y*, & *z*);

পরিগণক সম্পূর্ণ নির্দেশিকাটি পরীক্ষা ও অনুবাদের পরে (Compilation—যে ব্যাপারে একক 3 ..... আলোচনা করা হয়েছে) এই চলকগুলির মান জানতে চাইতে (cursor-টি সক্রিয় থাকবে অর্থাৎ পিট্ পিট্ (Blink) করবে), তখন এই চলকগুলির মান কীবোর্ডের মাধমে টাইপ করে দিতে হবে। মনে করা যাক এদের মানগুলি হলো $x = 10, y = 15{\cdot}25$ এবং z = 'A'। নীচের যে কোন একটি উপায়ে এই মানগুলি টাইপ করা যেতে পারে।

(i) 10 ␣ 15·25 ␣ A ↵ (দুটি মানের মধ্যে ␣ থাকবে যাতে মানগুলিকে আলাদা করে বোঝা যায়)

(ii) 10 ␣ 15·25 ↵
A ↵

(iii) 10 ↵
15·25 ␣ A ↵

(iv) 10 ↵
15·25 ↵
A ↵

(↵ এই চিহ্নটি 'Enter'—করা বোঝাচ্ছে)

এখানে প্রকৃতি নির্দেশক তন্ত্রী এবং চলকগুলির প্রকৃতির মধ্যে সামঞ্জস্য থাকতেই হবে, অন্যথায় 'Type mismatch'—ত্রুটি ঘটবে।

দ্রষ্টব্য ঃ তন্ত্রীচলকের ক্ষেত্রে scanf-এ চলকটির আগে &-চিহ্নটি বসানোর প্রয়োজন নেই। যেমন ধরা যাক্ $a = 20,\ b = 15{\cdot}75,\ c$ = 'M' এবং d = "ROLL"—এক্ষেত্রে scanf-টি হবে ঃ

scanf ("%*d* %*f* %*c* %*s*", &*a*, &*b*, &*c*, *d*) ;

আউটপুট অপেক্ষক printf-এর সাধারণ রূপ ঃ

এর সাধারণ রূপটি হলো

printf ("চলকগুলির প্রকৃতি নির্দেশক তন্ত্রী", চলক১, চলক২,.....); scanf-এর মতন printf-এ প্রকৃতি নির্দেশক তন্ত্রী একই প্রকার। তবে এখানে চলকগুলির Address নয়, সরাসরি চলকগুলির নাম দিতে হবে।

উদাহরণ (১) ঃ int $p = 4,\ q = 8$;

float $r = 3{\cdot}14159,\ s = 1{\cdot}762941$;

printf ("%f %d %f %d", *s*, *p*, *r*, *q*);

আউটপুট হবে ঃ

1·762941 4 3·14159 8

সাধারণভাবে দুটি মানের মধ্যে 2টি শূন্যস্থান থাকবে।

(২) ধরা যাক্ float $x = 17{\cdot}294176,\ y = -16{\cdot}214801$;

print (" %f %e", *x*, *y*);

-এর আউটপুট হবে ঃ

17·294176 – 0·162148e02

(আউটপুটের মান সম্বন্ধে কোন ধারণা না থাকলে e-format-ই ব্যবহার করা উচিৎ—সাধারণভাবে আউটপুটের নির্ভুলতা 6 অথবা 8 সার্থক অঙ্ক পর্যন্ত হয়)

(৩) অনেক সময় printf-এ কোন (message) লিখিত আকারে দেওয়া যায়। যেমন

int $x = 46$; (x = ছাত্রসংখ্যা)

float $y = 151{\cdot}25$; (y = গড় উচ্চতা সে. মি তে)

printf (”Number of student = % d Av-height = % cm”, x, y);

আউটপুট হবে ঃ

Number of students = 46 Av · height = 151·25 cm

(৪) অনেক সময় printf–অপেক্ষকে চলকের পরিবর্ত্তে কোন রাশিমালার উল্লেখ করা হয়, আউটপুটে ঐ রাশিমালার মান পাওয়া যাবে। যেমন

int $x = 4,\ y = 9$;

print (” %d %d $x + y = $%d”, $x + y$);

আউটপুট হবে ঃ $x + y = 13$

(৫) printf বিবৃতির মাধ্যমে আমরা যে আউটপুট পাই তার নির্মিতি পরিগণক কর্তৃক নির্দ্দিষ্ট। তবে ব্যবহারকারী তাঁর নিজের মতন করেও আউটপুট তৈরী করে নিতে পারেন। এজন্য printf অপেক্ষকে অতিরিক্ত কিছু নির্দেশ সন্নিবিষ্ট করা দরকার. এগুলি হলো আউটপুটগুলি কতটা জায়গা নিয়ে লেখা হবে, একই লাইনে লেখা হবে না কি পর পর লাইনে হবে ইত্যাদি।

প্রথমেই ধরা যাক প্রথম উদাহরণে s, p, r, q-কে একই লাইনে না লিখে, s ও p প্রথম লাইনে এবং r ও q কে পরের লাইনে লিখতে হবে। এর জন্য প্রয়োজনীয় নির্দেশটি হল $\backslash n$ অর্থাৎ নতুন লাইন শুরু করার নির্দেশ ঃ

printf (” %f %d $\backslash n$ %f %d”, s, p, r, q);

আউটপুট হবে 1·762941 4

3·14159 8

আউটপুটের গঠন বা নির্মিতি নিজের মতন করে করতে হলে printf—বিবৃতিতে প্রতিটি আউটপুট কটা ঘর অধিকার করে থাকবে তার একটা নির্দেশ দিতে হবে। এই নির্দেশগুলি নীচের উদাহরণের সাহায্যে দেখানো হল ঃ

মনে করা যাক দেওয়া আছে int $x = 7527$; এটিকে সঠিক ও সম্পূর্ণভাবে লিখতে 4-টি ঘরের (space) প্রয়োজন।

(i) printf (” % d ”, x) :

7	5	2	7

(ii) printf (”% 4 d ”, x) :

7	5	2	7

(iii) printf (” % 6 d”, x) :

		7	5	2	7

(% 6d —এই নির্দেশের মাধ্যমে x-এর মানকে লেখার জন্য ছয়টি জায়গা (space) অধিকার করে রাখা হয়েছিল, কিন্তু প্রয়োজন ছিল 4-টি জায়গার, অতিরিক্ত দুটি ঘর বাঁদিকে খালি রয়ে গেল)

(iv) printf (" % 2*d* ", *x*) : | 7 | 5 | 2 | 7 |

(% 2*d* মাধ্যমে *x*-এর জন্য দুটিমাত্র জায়গা রাখা হয়েছিল, কিনতু প্রয়োজন ছিল অন্ততঃ 4-টি জায়গার, এক্ষেত্রে পরিগণক প্রদত্ত নির্দেশটি অগ্রাহ্য করে ন্যূনতম যে কয়টি ঘরের প্রয়োজন (এখানে 4টি ঘর) ততটা জায়গাই সে লেখার জন্য নেবে)

(v) printf (" % - 7*d* ", *x*) : | 7 | 5 | 2 | 7 | | | |

(– চিহ্নটি আউটপুটের লেখাটিকে বাঁদিকে সরিয়ে আনবে, 7-টি ঘরের বাঁদিকের প্রথম 4-টি ঘর লেখার জন্য প্রয়োজন হবে, ডানদিকের অবশিষ্ট তিনটি ঘর খালি থাকবে)

তুলনীয় ঃ printf (" %7d ", x) : | | | | 7 | 5 | 2 | 7 |

(ডানদিকে সরে লেখার জন্য)

(vi) printf (" % 06*d* ", *x*) : | 0 | 0 | 7 | 5 | 2 | 7 |

('0' চিহ্নটি মোট ছয়টি ঘরের বাঁদিকের প্রথম দুটি ঘরে 0 লিখেবে অর্থাৎ " % 6*d* "-তে বাঁদিকে যে দুটি শূন্যস্থান ছিল, সেগুলি 0 দ্বারা প্রতিস্থাপিত হবে)

বাস্তব অর্থাৎ Floating Point সংখ্যাকে লেখার জন্য আউটপুটের বিভিন্ন প্রকারভেদ ঃ

উদাহরণ হিসেবে নেওয়া হল ঃ float *x* = 27·8729 ;

(i) printf (" %*f*", *x*) : | 2 | 7 | · | 8 | 7 | 2 | 9 |

(% *f* -এই নির্দেশে আউটপুটটি ঠিক যতগুলি ঘর প্রয়োজন ততগুলি ঘরই ব্যবহার করবে)

(ii) % 7·4*f* : এখানে *x*-এর প্রদত্ত মানটিকে লেখার জন্য এই নির্দেশটিই যথাযথ কারণ এই নির্দেশটির অর্থ হলো মোট 7-টি ঘরের প্রয়োজন (· চিহ্নটি ধরে) যার মধ্যে 4-টি ঘর দশমিকের পরে থাকবে।

(iii) % 9·5*f* : Print (" % 9·5*f*", *x*) : | 2 | 7 | · | 8 | 7 | 2 | 9 | 0 |

এই % 9·5*f* আকারটি লক্ষ্য করা যক্। এর মাধ্যমে বলা হয়েছে *x*-রে মানকে লেখার জন্য মোট 9-টি ঘর নির্দ্দিষ্ট কর যার মধ্যে দশমিক বিন্দুর পরে 5-টি ঘর নির্দ্দিষ্ট থাকবে। লক্ষ্যণীয় যে *x*-এর প্রদত্ত মানটিকে লেখার জন্য যথার্থ আকারটি হল % 7·4 *f* কারণ *x*-এর এই মানটিকে মোট 7-টি বর্ণ আছে, (দশমিক বিন্দু সহ) যার মধ্যে 4-টি বর্ণ দশমিক বিন্দুর পরে আছে। যেহেতু *x*-এর প্রদত্ত মানটি ধনাত্মক অতএব তার পূর্বে + চিহ্নটি দেওয়া আবশ্যিক নয়। কিন্তু *x* = –27·8729 হলে যথার্থ আকারটি হতো %8·4 *f* :

| – | 2 | 7 | · | 8 | 7 | 2 | 9 |

(iv) % 7·3 *f* : printf (" % 7·3*f*", *x*) | 2 | 7 | · | 8 | 7 | 3 |

উল্লেখিত আকরটিতে দশমিক বিন্দুর পরে 3টি ঘর নির্দ্দিষ্ট করা হয়েছে—এই কারণে 4—দশমিকের সংখ্যাটির তিন দশমিক স্থান পর্যন্ত আসন্ন মান লেখা হলো।

অনুরূপভাবে % 7·0 *f* : printf (" % 7·0*f* ", *x*) : | | | | | | 2 | 8 |

(দশমিকের পরে কোন ঘর নির্দ্দিষ্ট করা হয় নি এই কারণে x-এর নিকটস্থ পূর্ণসাংখিক মানটিই (= 28) লেখা হবে—বাকী 5-টি বাঁদিকের ঘর খালি থাকবে)

(v) % · 0f : printf (" ·0f ", x) : | 2 | 8 |

(% · 0f -এর আকার-নির্দেশে x-এর নিকটস্থ পূর্ণসাংখ্যিক মানটি লেখা হবে)

(vi) % · 2f : praintf (" % · f ", x) : | 2 | 7 | · | 8 | 7 |

(দশমিকের পরে দুই ঘর পর্যন্ত আসন্ন মান লেখা হবে, এর জন্য ঠিক যতগুলি ঘর প্রয়োজন, ততগুলি ঘরই লেখার জন্য নেবে)

(vii) % – 7·3 f : printf (" % –7·3f", x) : | 2 | 7 | · | 8 | 7 | 3 | |

('–' চিহ্নটির জন্য বাঁদিক থেকে লেখা শুরু হবে)

(viii) % 07 · 3f : | 0 | 2 | 7 | · | 8 | 7 | 3 |

(ix) % 12·6 e : printf (" % 12·6 e ", x) : | 0 | · | 2 | 7 | 8 | 7 | 2 | 9 | e | + | 0 | 2 |

এই প্রকার আকার-নির্দেশ বিশেষ ভাবে লক্ষ্যণীয়। এর সাধারণরূপ হল % w · de (উপরের উদাহরণে $w = 12, d = 6$) এখানে, w হলো মোট যতগুলি ঘর নিয়ে লেখা হবে, সেই ঘর সংখ্যার নির্দেশ এবং d হল দশমিক বিন্দুটির পরে যতগুলি ঘর ব্যবহার করা হবে তার সংখ্যা। এছাড়াও লেখাটিতে আরও কয়েকটি বর্ণ আছে, যেমন : e, + (অথবা –) e-র পরে সূচক-নির্দেশক দুটি সংখ্যা, দশমিক বিন্দু, দশমিক বিন্দুর পূর্বে 0 সংখ্যাটি এবং ঋণাত্মক সংখ্যার ক্ষেত্রে সবার প্রথমেই—চিহ্নটি। অর্থাৎ মোট 7-টি চিহ্ন/বর্ণ (ধনাত্মক সংখ্যার ক্ষেত্রে অবশ্য 6-টি), অতএব w অবশ্যই (d+7) বা তার থেকে বেশী হবে।

অর্থাৎ $w \geq d + 7$ (ধনাত্মক সংখ্যার ক্ষেত্রে $w \geq d + 6$) এই কারণে % 10·6e এই আকার-টি গ্রহণযোগ্য নয় কারণ এখানে $w = 10, d = 6$ যা $w \geq d + 7$ (অথবা $w \geq d + 6$) এই শর্তটি পূরণ করে না।

(x) % 14·7e : | | 0 | · | 2 | 7 | 8 | 8 | 7 | 2 | 9 | 0 | e | + | 0 | 2 |

(xi) % – 14·3e : | 0 | · | 2 | 7 | 9 | e | + | 0 | 2 | | | | | |

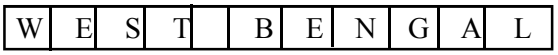
তন্ত্রীচলক লেখার জন্য প্রয়োজনীয় আউটপুটের বিভিন্ন আকার ঃ

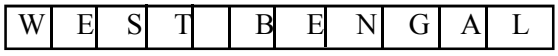
এর সাধারণ রূপটি হলো ঃ % w · ds যেখানে w হলো মোট ঘরের সংখ্যা এবং d হলো ঐ তন্ত্রীচলকের প্রথম যে কটি বর্ণ লেখা হবে তার সংখ্যা। উদাহরণ হিসেবে একটি তন্ত্রী state নেওয়া হলো ঃ

char state [ ] = "WEST ␣ BENGAL" ;

এই তন্ত্রীটিতে মোট 11-টি বর্ণ আছে (একটি ␣ সমেত), অতএব এটি লেখার জন্য যথার্থ আকারটি হলো " % 11s",। এবার আমরা দেখবো বিভিন্ন আকার-নির্দেশে কি ধরনের আউটপুট পাওয়া যাবে ঃ

(i) printf (" % s", state) : | W | E | S | T | | B | E | N | G | A | L |

(ii) printf (" % 8s ", state) : 

(iii) printf (" % 11s ", state) : 

(iv) printf (" % 14s", state) :

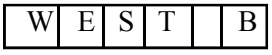
দ্রষ্টব্য ঃ প্রথম দুটি উদাহরণ থেকে এটা বোঝা গেল যে প্রয়োজনের তুলনায় কম ঘর নির্দ্দিষ্ট করলেও পরিগণক তা অগ্রাহ্য করে সম্পূর্ণ তন্ত্রীটিকেই লিখবে, ঠিক যতগুলি ঘর প্রয়োজন ততগুলিই সে ব্যবহার করবে। প্রয়োজনের তুলনায় বেশী ঘর (উদাহরণ (iv)) নির্দ্দিষ্ট করলে বাঁদিকে অতিরিক্ত ঘরগুলি খালিই থাকবে।

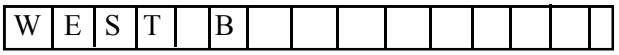
(v) printf (" % – 15*s* ", state) :

('–' চিহ্নের সাহায্যে লেখাটিকে বাঁদিকে সরিয়ে আনা হলো, এবং অতিরিক্ত ঘরগুলি ডানদিকে খালি থাকবে)

(vi) printf (" % 15·6*s*", state)

(15·6s : মোট ঘরের সংখ্যা = 15, লিখতে হবে প্রথম 6-টি বর্ণ—ডানদিক বরাবর; অতিরিক্ত ঘরগুলি বাঁদিকে খালি থাকবে)

(vii) printf (" % ·6s", state) : 

(viii) printf (" % – 15·6s", state) : 

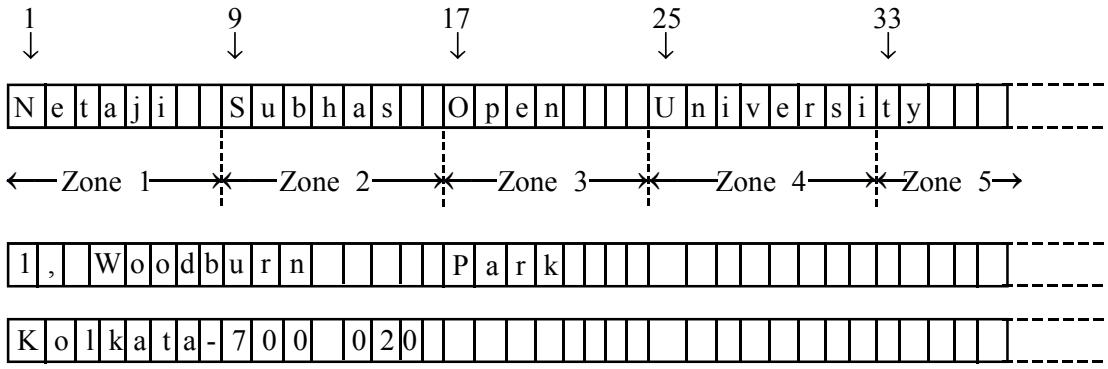
কোন আউটপুটকে সুসংবদ্ধ আকারে প্রকাশ করতে হলে Printf ( ) এই ক্রিয়াটি আউটপুটের আকার নিরূপক কিছু নির্দেশ দেওয়া হয়, যার মধ্যে বহুল ব্যবহৃত কিছু নির্দেশ নিয়ে উপরে আলোচনা করা হলো। এছাড়াও আরো কিছু চিহ্ন আছে যার সাহায্যে আউটপুটচিকে ভেঙ্গে ভেঙ্গে বিভিন্ন লাইনে লেখা যেতে পারে। এগুলিকেই Escape Sequece বা নির্গমন সূচক চিহ্ন বলা যেতে পারে। বহুল ব্যবহৃত কয়েকটি নির্গমন সূচক চিহ্ন হলো \ t, \ n, \ b, ইত্যাদি। এদের ব্যবহারগুলি নীচের কয়েকটি উদাহরণের মাধ্যমে দেখানো হলো।

\ t : এর ব্যবহার BASIC ভাষার TAB-ক্রিয়াটির মতন। সাধারণ ভাবে 80 কলামের Monitor-এর পর্দায় আমরা আউটপুটটিতে দেখতে পাই। \ t-এর সাহায্যে এই দৃশ্যমান অঞ্চলটি সাধারণতঃ 10-টি জোন (Zone)-এ বিভক্ত করা হয়, Zone 1, Zone 2, ... Zone 10, প্রতিটি Zone সাধারণত 8-column-এর হয়ে থাকে। \t -এই চিহ্নের মাধ্যমে এক জোন থেকে পরবর্তী আর এক জোন-এ লেখাটিকে সরানো যায়।

\ n এই চিহ্নের মাধ্যমে আউটপুটটিকে নতুন লাইন থেকে লেখানো হয়।

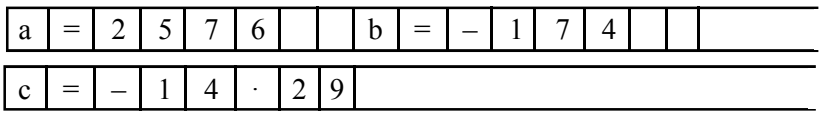
উদাহরণ ১ ঃ printf (" Netaji\ t Subhas\ t Open\ t University\ n 1, Woodburn\ t Park\ n Kolkata-700 020") ;

আউটপুটটি হবে ঃ



উদাহরণ (২) ঃ int a = 2576, b = – 174;
float = –14·29;
printf (" a = % d \ t b = % d\ n c = % f ", a, b, c);

আউটপুটটি হবে ঃ 1 9



(\ t চিহ্নটি ব্যবহার না করলে পরপর দুটি মানের মধ্যে সাধারণভাবে দুটি ঘরের ফাঁক থাকে।)

C–ভাষার কার্য পর্যায়ক্রম নিয়ন্ত্রণকারী নির্দেশাবলী (Control Statement)

কয়েকটি ছোট C–নির্দেশাবলী নিয়ে এই অংশের আলোচনা শুরু করা যাক।

C–নির্দেশাবলী (১) ঃ

```
# include < stdio · h >
main ( )
{
int  x = 12,  y = 3 ;
float  z = 12 · 9 ,  a ;
a = (x + y) * z ;
printf  ( " a = % f  ",  a ) ;
}
```

আউটপুট ঃ $a = 193 \cdot 5$

C–নির্দেশাবলী (২) ঃ

```
/* To find the area of a triangle */
    # include < stdio · h >
    # include < math · h >
    main ( )
    {
    float a = 5·1, b = 7·3, c = 11·5, s, area ;
    s = (a + b + c) / 2·0;
    area = sqrt ((s − a) * (s − b) * (s − c) * s);
    printf ( " AREA = %f ", area) ;
    }
```

এই ধরণের নির্দেশাবলী কার্যকরী করতে পরিগণক একটি পর্যায়ক্রম তৈরী করবে—এক্ষেত্রে এই পর্যায়ক্রমগুলি হবে স্বাভাবিক পর্যায়ক্রম, অর্থাৎ পরপর উপর থেকে নীচে—যে ক্রমানুসারে নির্দেশগুলি দেওয়া আছে, সেই অনুসারে। কিন্তু বহু ক্ষেত্রেই এই স্বাভাবিক ক্রম অনুসারে নির্দেশগুলি কার্যকরী করা সম্ভব নয়, কারণ অনেক ক্ষেত্রে নির্দেশিকার কিছু কিছু অংশ চলকের বিভিন্ন মানের জন্য বার বার করতে হতে পারে অথবা বিশেষ কোন শর্তসাপেক্ষে নির্দেশিকার এক অংশ থেকে সরাসরি অন্য কোন অংশে যেতে হতে পারে—এই রকম ক্ষেত্রে স্বাভাবিক পর্যায়ক্রমের পরিবর্তন করতে হয়।

C–ভাষাতে পর্যায়ক্রম নিয়ন্ত্রণকারী নির্দেশগুলি হলো ঃ

(i) যৌক্তিক IF নির্দেশমূলক বিবৃতি (Logical IF Statement)

(ii) IF–ELSE বিবৃতি

(iii) শর্তনিরপেক্ষ GOTO বিবৃতি (Uncondifional goto)

(iv) Switch বিবৃতি

এই বিবৃতিগুলির সঠিক আকার ও তাদের ব্যবহার সম্বন্ধে নীচে আলোচনা করা হলো।

(১) যৌক্তিক IF (Logical IF) ঃ এই নির্দেশমূলক বিবৃতিটির সাধারণ রূপ হলো।

```
    if (শর্ত)
    statement 1 ;
```

ঐ শর্তটি পূরণ হলে তবেই statement 1-এর নির্দেশটি কার্যকর হবে, অন্যথায়, অর্থাৎ শর্তটি পূরণ না হলে, statement 1 এর পরবর্তী নির্দেশের কাজ শুরু হবে। if—বিবৃতির শেষে কোন (;) বসবে না।

উদাহরণ (1) if $(b*b-4*a*c>=0)$

printf ("Roots are real" $\backslash n$) ;

(2) if $(x==a)$ $y=\text{pow}$ $(x,2)$;

প্রথম উদাহরণে $b^2-4ac\geq 0$ হলে Roots are real এই বাক্যটি লেখা হবে এবং দ্বিতীয় উদাহরণে x এবং a-র মান সমান হলে তবেই $y=x^2$ হবে।

if—বিবৃতিটিতে একের অধিক নির্দেশমূলক বিবৃতি থাকলে সম্পূর্ণ if—ব্লকটিকে একজোড়া ধনুর্বন্ধনী { }-র মধ্যে রাখতে হয় যেমন ঃ

```
if (b*b-4*a*c>=0)
{
    x1 = (-b+sqrt(b*b-4*a*c)) / (2*a);
    x2 = (-b-sqrt(b*b-4·*a*c)) / (2·*a);
    printf (" Roots are real : %f %f", x1, x2);
}
```

if-এর ব্লক

(২) if-else বিবৃতি ঃ এটি if—বিবৃতির একটি বৃহত্তর রূপ। if-এর পরে যে শর্তটি থাকবে সেটি পূরণ হলে if-ব্লকের অধীনস্থ নির্দেশগুলি পালিত হবে, না হলে else-ব্লকের অধীনস্থ নির্দেশগুলি কার্যকরী হবে।

সাধারণরূপ ঃ

```
if (শর্ত)
{
    s1 ;
    s2 ;
    ⋮
    sk ;
}
else
{
    s'1 ;
    s'2 ;
    ⋮
    s'm ;
}
```

if-ব্লক (শর্তটি পূরণ হলে)

else-ব্লক (শর্তটি পূরণ না হলে)

উদাহরণ (১) ঃ float $a, b, c, d,\ x_1, x_2$;

$d = b*b - 4\cdot 0*a*c$;

if $(d >= 0)$

{

$x_1 = (-b + \text{sqrt}(d)) / (2\cdot 0*a)$;

$x_2 = (-b - \text{sqrt}(d)) / (2\cdot 0*a)$;

printf (" Roots are real : %f f", x_1, x_2) ;

}

else

printf (" Roots are imaginary ") ;

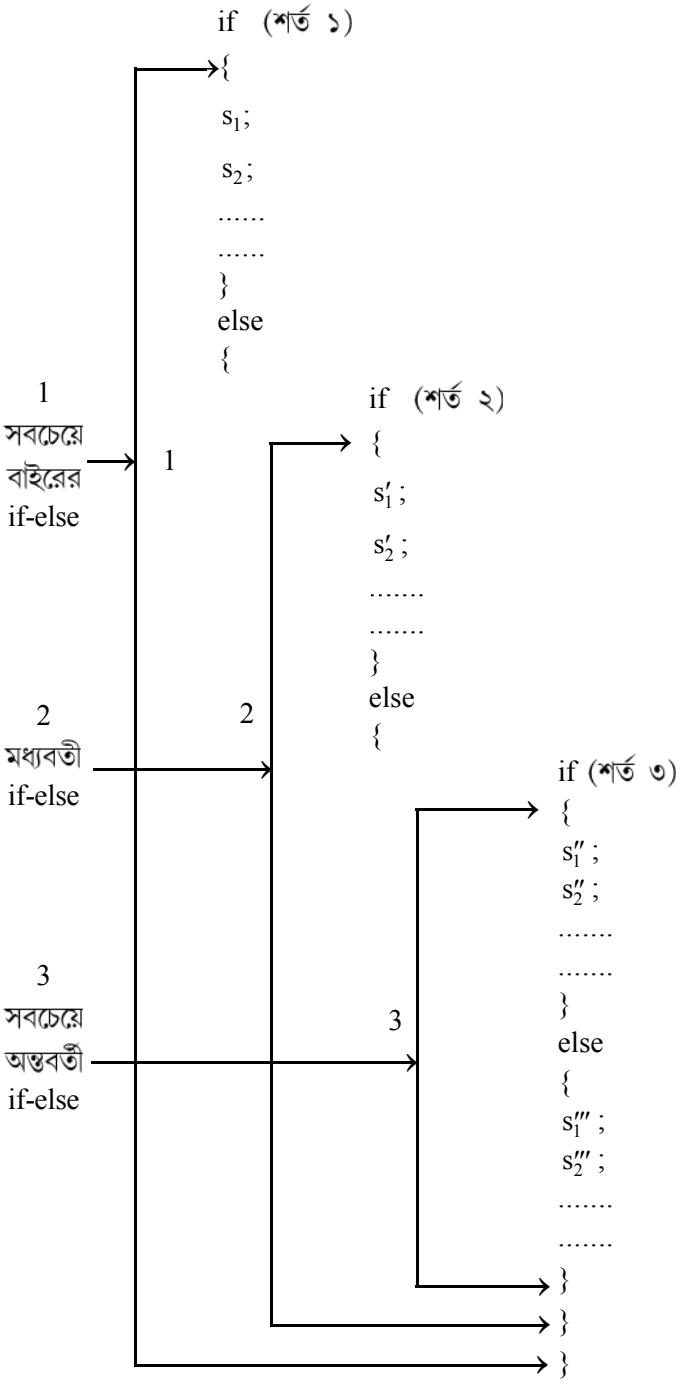
উদাহরণ (২) ঃ মনে করা যাক x-চলকের একটি অপেক্ষক $f(x)$ এরূপভাবে সংজ্ঞায়িত :

$$f(x) = 1 + \cos x, \quad \text{যদি } x > 2\cdot 0 \text{ হয়}$$
$$= \log(x^2 + 2), \quad \text{যদি } x = 2\cdot 0 \text{ হয়}$$
$$= 1 - e^x, \quad \text{যদি } x < 2\cdot 0 \text{ হয়}$$

C–নির্দেশের সাহায্যে $f(x)$ কে প্রকাশ করতে হলে আমাদের if-else নির্দেশমূলক বিবৃতির মাধ্যমে লিখতে হবে ঃ

```
float x, f
if (x = = 2·0)
f = log (x * x + 2·0);
else
{
if (x < 2·0)
f = 1 − exp (x);
}
else
f = 1 + cos (x);
}
```

এই উদাহরণের বিশেষ দিকটি হলো if-else বিবৃতির অন্তর্গত আর একটি if-else উপস্থিতি। একে বলা হয় nested if-else. এই ধরণের nested if-else-এর সাধারণ একটি রূপ হলো



(৩) শর্তনিরপেক্ষ goto বিবৃতি

goto—এই নির্দেশের সাহায্যে পর্যায়ক্রম নিয়ন্ত্রককে যে কোন পর্যায়ে আটকে দিয়ে নির্দেশাবলীর অন্য কোন অংশে (পূর্ববর্তী বা অগ্রবর্তী) নিয়ে যাওয়া যায়। goto সাধারণ রূপটি হলো ঃ

goto < lebel > ;

যেমন goto s_1 ;

goto s_{10} ;

ইত্যাদি

যেখানে s_1 বা s_{10} সমগ্র নির্দেশাবলীর অন্তর্গত কোন বিশেষ নির্দেশ যাকে s_1 বা s_{10} দ্বারা চিহ্নিত করা হয়েছে। got_0 s_1 নির্দেশ দ্বারা নিয়ন্ত্রককে বর্তমান স্থান থেকে s_1 দ্বারা চিহ্নিত নির্দেশটিতে পাঠানো হয়েছে।

কোন বিবৃতিকে কিভাবে চিহ্নিত করা হয় ঃ সাধারণভাবে C–ভাষায় লেখা কোন নির্দেশের বিবৃতিগুলি কোন রকম সংখ্যা বা নাম দ্বারা চিহ্নিত করা হয় না। তবে goto বিবৃতি ব্যবহার করলে বর্তমান ক্রম থেকে যে ক্রমে নিয়ণন্ত্রককে নিয়ে যাওয়া হবে সেই বিবৃতিটিকে নামকরণ বা চিহ্নিতকরণের (lebelling) প্রয়োজন আছে। এই চিহ্নিতকরণ যে কোন চলকের মতন নাম দ্বারা করা হয়ে থাকে। যেমন

start :

statement 10 ;

read :

statement 3 ;

অথবা s10 : statement 5 ;

end : statement 20 ;

এই চিহ্নিতকরণের মাধ্যমে statement 10 কে start, statement 3 কে read, statement 5 কে s10 দ্বারা নামাঙ্কিত করা হয়েছে।

এ প্রসঙ্গে কয়েকটি বিষয় মনে রাখা দরকার ঃ

(১) প্রতিটি goto-বিবৃতিতে একটি মাত্র নির্দেশের প্রতি চিহ্নিত থাকবে।

(২) বিভিন্ন goto-বিবৃতি একই নির্দেশের প্রতি চিহ্নিত হতে পারে

(৩) বিবৃতির নামকরণকারী চিহ্নটি ঐ নির্দেশাবলীতে কোন চলক হিসাবে ব্যবহার করা যাবে না।

(৪) একই C–নির্দেশাবলীতে একই নামযুক্ত একাধিক বিবৃতি থাকবে না।

C–ভাষাতে goto-র ব্যবহার ঃ এরজন্য if-else-এর উদাহরণ (২)-এর অপেক্ষকটি নেওয়া হলো। goto-র সাহায্যেও $f(x)$ সংজ্ঞায়িত করা যেতে পারে ঃ

float x, f ;

if $(x == 2 \cdot 0)$

goto $s1$;
else
{
if $(x < 2 \cdot 0)$
goto $s2$;
}
else
$f = 1 + \cos(x)$

$s1 : f = \log(x * x + 2 \cdot 0)$

$s_2 : f = 1 - \exp(x)$;

বিঃ দ্রঃ C–ভাষাতে goto-র ব্যবহার যথেষ্ট প্রচলিত নয়।

(4) switch—প্রক্রিয়া নির্দেশক বিবৃতি ঃ

C–ভাষাতে switch বিবৃতিটি নির্দেশাবলীটিকে নানা প্রশাখায় বিভক্ত করে। এটিকে if-else বিবৃতিটির একটি বৃহত্তর সংস্করণ বলে মনে করা যেতে পারে। if-else নির্দেশে আমরা দেখেছি কয়েকটি শর্ত পূরণ হওয়ার সাপেক্ষে নির্দেশাবলীর কোন কোন অংশের প্রক্রিয়াগুলি সম্পন্ন হয় এবং শর্তপুরণ না হওয়ার কারণে নির্দেশাবলীর অন্য কোন কোন অংশের প্রক্রিয়াকরণের কাজটি স্থগিত রাখা হয়। শর্তগুলির সংখ্যা বেশী হলে if-else-এর সাহায্যে নির্দেশগুলি যথাযথভাবে সন্নিবিষ্ট করা যথেষ্ট কঠিন এবং অনেক সময় বিভিন্ন বিকল্পগুলির (else-দ্বারা নির্দেশিত) মধ্যে যোগসূত্রটি সঠিকভাবে বজায় রাখার কাজটিও বেশ জটিল হয়ে পড়ে। এরকম ক্ষেত্রগুলিতে অনেক সময় (switch-break)-নির্দেশটি যথেষ্ট কার্যকরী হয়। (swith-break)-নির্দেশটির সাধারণ কাঠামোটি হলো ঃ

switch কাঠামোর শুরু → switch (পূর্ণসাংখ্যিক রাশিমালা)
{
case n_1 :
statement ;
statement ;
.....
statement ;
break ;
case n_2 :
statement ;
statement ;
.....
statement ;

ব্লক-১

ব্লক-২

```
        break ;
    .........
    ..........
    case  n_k  ;
                statement ;                 ↑
                statement ;                 |        ব্লক-K
                ..............              |
                statement ;                 ↓
        break ;
    default :
                statement ;                 ↑
                statement :                 |        ব্লক-Default
                ................            |
                statement ;                 ↓
        }            ←—— switch-কাঠামোর শেষ
        statement x ;
```

এটি কেমনভাবে কাজ করে?

প্রথমেই পূর্ণসাংখিক রাশিমালাটির মান নির্ণয় করা হয়। এটি অবশ্যই একটি পূর্ণসংখ্যা হবে। এই মানটি যদি n_1-এর সমান হয়, তবে case n_1-এর অন্তর্গত ব্লক-১-এর প্রক্রিয়াগুলি সম্পন্ন হবে, এবং ব্লকের শেষে break -নির্দেশের মাধ্যমে প্রক্রিয়া নিয়ন্ত্রক switch-কাঠামোটির বাইরে statement x-এ চলে যাবে এবং সেখান থেকে পরবর্তী নির্দেশ অনুসারে প্রক্রিয়াগুলি সম্পন্ন করবে। অনুরূপভাবে ঐ পূর্ণসংখ্যাটির মান n_2, n_3, ..., বা n_k হলে, সেই অনুযায়ী নির্দিষ্ট ব্লক-টির কাজগুলি সম্পন্ন হবে এবং নিয়ন্ত্রকটি switch-কাঠামোর বাইরে চলে যাবে। কিন্তু পূর্ণসংখ্যাটির মান n_1, n_2, ... বা n_k-র কোনটাই যদি না হয়, তবে সেক্ষেত্রে default-অংশের কাজটি সম্পন্ন হবে। এখানে লক্ষ্যণীয় default ব্লকের শেষে কোন break নেই কারণ এক্ষেত্রে স্বাভাবিকভাবেই নিয়ন্ত্রকটি switch-কাঠামোটির বাইরে চলে আসবে।

break-এর কাজ কী? ঃ break-নির্দেশটি প্রক্রিয়া সম্পাদনের কাজটি শুধুমাত্র নির্দিষ্ট ঐ ব্লকটির মধ্যেই সীমাবদ্ধ রাখে এবং কাজ শেষ হলে নিয়ন্ত্রকটি সরাসরি switch-কাঠামোর বাইরে চলে যায়। অন্যান্য ব্লক-গুলির নির্দেশগুলিতে কোনরূপ হস্তক্ষেপ হয় না। break-নির্দেশটি না থাকলে কোন ব্লকের কাজ শেষ করে নিয়ন্ত্রকটি পরবর্তী ব্লকের প্রক্রিয়াগুলি কার্যকর করতে শুরু করতো।

switch–কাঠামোর ব্যবহার ঃ

ধরা যাক্ $f(x)$-একটি অপেক্ষক, নিরপেক্ষ চলক x-এর বিভিন্ন পূর্ণসাংখ্যিক মানের জন্য এইভাবে সংজ্ঞাত ঃ

$f(x) = 2x^2$ যখন $x = 1, 3, 5$

$= 4x + 3$ যখন $x = 2$

$= \cos x + \sin x$ যখন $x = 4$ এবং 6

$= e^x$ যখন $x > 6$

এই ধরনের অপেক্ষকের জন্য if-else-এর ব্যবহার আগেই দেখানো হয়েছে। এবার switch-কাঠামোর দ্বারা এই অপেক্ষকটিকে আমরা সংজ্ঞাত করবো ঃ

```
switch (x)
{
case 1 :
        f = 2*x*x ;
        break ;
case 2 :
        f = 4.0*x+3;
        break ;
case 3 :
        f = 2*x*x;
        break ;
case 4 :
        f = cos(x)+sin(x);
        break ;
case 5 :
        f = 2*x*x;
break ;
case 6 :
        f = cos(x)+sin(x);
break ;
default :
        f = exp(x);
}
```

C–ভাষাতে লূপের (Loop) ব্যবহার

কোন নির্দেশিকার বিশেষ কোন অংশকে অনেক ক্ষেত্রেই বারবার সম্পন্ন করতে হয়। সেক্ষেত্রে লূপের ব্যবহার খুবই কার্যকরী হয়। একটি উদাহরণ দিয়ে ব্যাখ্যা করা যাক্ ঃ

মনে করা যাক কোন গাণিতিক প্রক্রিয়ায় আমরা sum-এই চলকটির মান নির্ণয় করবো যেখানে

$$\text{sum} = 51^3 + 53^3 + 55^3 + ... + 99^3$$

এই প্রক্রিয়াটি সম্পন্ন করার জন্য নীচের কার্য পদ্ধতিটি (algorithm) অনুসরণ করা যেতে পারে ঃ

প্রথম পদক্ষেপ ঃ sum-এর প্রাথমিক মান = 0 ধরা হলো

দ্বিতীয় পদক্ষেপ ঃ $n = 51$ ধরা হলো

তৃতীয় পদক্ষেপ ঃ n^3-এর মান নির্ণয় করা হলো

চতুর্থ পদক্ষেপ ঃ n^3-এর প্রাপ্ত মানটি sum-এর মানের সঙ্গে যোগ করা হলো এবং sum-এর বর্তমান মানের সাহায্যে পূর্ববর্তী মানটি প্রতিস্থাপিত করা হলো।

পঞ্চম পদক্ষেপ ঃ n-এর মান 2 বাড়ানো হলো অর্থাৎ $n = n + 2$ করা হলো।

ষষ্ঠ পদক্ষেপ ঃ যদি $n \leq 99$ হয় তবে তৃতীয়, চতুর্থ ও পঞ্চম পদক্ষেকগুলি পুনরায় করা হলো। অন্যথায় প্রক্রিয়াটি শেষ করা হল।

শেষ পদক্ষেপ ঃ পঞ্চম পদক্ষেপের শেষে প্রাপ্ত sum-এর মানই হলো আমাদের উদ্দিষ্ট মান।

গণিতের ভাষায় ঃ

(1) sum = 0 (প্রাথমিক স্তর)

(2) sum = $\underline{0}$ + 51^3 (১ বার)

(3) sum = $\underline{(0 + 51^3)}$ + 53^3 (২ বার)

(4) sum = $\underline{(0 + 51^3 + 53^3)}$ + 55^3 (৩ বার)

.....

(26) sum = $\underline{(0 + 51^3 + 53^3 + ... + 97^3)}$ + 99^3 (২৫ বার)

বন্ধনীতে আবদ্ধ এবং রেখাঙ্কিত অংশটি হল ঐ স্তরের sum-এর পূর্ববর্তী মান।

C–ভাষাতে এই নির্দেশগুলি এইভাবে লেখা যেতে পারে ঃ

```
        int n, sum ;
        sum = 0 ;
        n = 51 ;
s1 : sum = sum + pow (n, 3) ;
        if (n = = 99)
        goto print ;
        else
        {n = n + 2 ;
         goto s1 ;}
print : printf (" SUM = % d ", sum);
```

নির্দেশিকাটি ভালোভাবে পরীক্ষা করলে দেখা যাবে যে

s1 : sum + pow (n, 3)-এই নির্দেশটি 25 বার কার্যকরী হয়েছে, অর্থাৎ এখানে একটি লূপ-কাঠামো ব্যবহার করা হয়েছে এবং এই লূপটি 25 বার আবর্তিত হয়েছে।

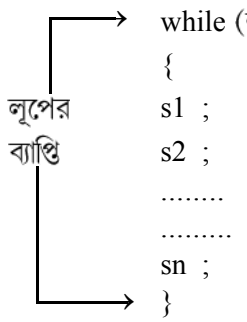
C–ভাষাতে লূপ-কাঠামো তৈরীর জন্য প্রধানতঃ তিন প্রকারের লূপ-বিবৃতি আছে ঃ

(১) while–বিবৃতি

(২) do-while বিবৃতি

(৩) for বিবৃতি

while – লূপ ঃ এই লূপের সাধারণ কাঠামোটি হলো ঃ




```
while (শর্ত)
{
s1 ;
s2 ;
.....
.....
sn ;
}
```

(দ্র ঃ লূপে একটিমাত্র বিবৃতি থাকলে { } ব্যবহার নাও করা যেতে পারে)

এটি কিভাবে কাজ করে? ঃ প্রথমেই শর্তটিকে পরীক্ষা করা হয়, সেটি T অথবা F হতে পারে। T-হলে, অর্থাৎ শর্তটি সিদ্ধ হলে, $s1, s2, ..., sn$ নির্দেশগুলি দ্বারা নির্ধারিত প্রক্রিয়াগুলি সম্পন্ন হবে এবং কার্য নিয়ন্ত্রকটি পুনরায় while-এ ফিরে আসবে এবং নতুন পরিস্থিতিতে নতুনভাবে শর্তটি পরীক্ষা করবে। এবারেও যদি শর্তটি সিদ্ধ হয় (অর্থাৎ True হয়) তবে আবার $s1, s2, ..., sn$-এর প্রক্রিয়াগুলি সম্পন্ন হবে এবং আবার নিয়ন্ত্রকটি while-এ ফিরে আসবে এবং শর্তটি পরীক্ষা করবে। এইভাবে যতক্ষণ পর্যন্ত শর্তটি সিদ্ধ হবে, ততক্ষণ পর্যন্ত লূপটি ক্রিয়াশীল থাকবে। যখনই শর্তটি অসত্য (False) হবে, তখনই নিয়ন্ত্রক লূপের কাজ শেষ করে পরবর্তী অংশে যাবে। একেবারে গোড়াতেই যদি শর্তটি অসিদ্ধ হয়, তবে while-লূপটি একবারও কার্যকরী হবে না।

উদাহরণ ঃ আগের উদাহরণটি while লূপের সাহায্যে করা যেতে পারে

```
int n, sum ;
n = 51
sum = 0 ;
while (n < = 99)
{
sum = sum + paw (n, 3) ;
n = n + 2 ;
}
print f (" sum = % d", sum)
```

দ্র ঃ (১) while-এর শেষে ; থাকবে না

(২) while-লূপে একাধিক বিবৃতি থাকলে লূপটিকে

{ } দ্বারা আবদ্ধ করা আবশ্যিক। অন্যথায় কেবলমাত্র while-এর পরবর্তী প্রথম বিবৃতির নির্দেশটিই শুধুমাত্র সম্পন্ন হবে

while-লূপের শর্তটি যৌক্তিক সম্পর্কও হতে পারে, যেমন ঃ

while ((aggregate > = 60 && math > = 80) && (eng > 40 || beng > = 40))

উপরের জটিল যৌক্তিক রাশিটি true হবে তখনই, যখন কোন পরীক্ষার্থী গণিতে 80% বা তার বেশী এবং সামগ্রিকভাবে 60% বা তার বেশী নম্বর পাবে এবং ইংরাজীতে 40% -এর বেশী অথবা বাংলাতে 40% বা তার বেশী নম্বর পাবে।

while-লূপ তৈরীর সময় একটি অত্যন্ত গুরুত্বপূর্ণ বিষয় মনে রাখতে হবে। লূপের শর্তটি এমন যেন না হয় যাতে শর্তটি সর্বদাই সিদ্ধ হয়, কখনই অসত্য হলো না—সেক্ষেত্রে লূপটি অসীম সংখ্যক বার কার্যকরী হতেই থাকবে, কখনই শেষ হবে না। নীচের উদাহরণটি পরীক্ষা করলে দেখা যাবে এটি আসলে একটি অসীম লূপ, কখনই থামবে না—

```
sum = 0 ;
n = 1 ;
while (n < = 10)
{
sum = sum + n * n ;
n = n – 1 ;
}
print f (” sum = % d” sum) ;
```

এখানে n-এর প্রাথমিক মান হলো 1, পরবর্তী মানগুলি হলো 0, –1, –2, –3, ... অর্থাৎ (n < = 10) -এই শর্তটি সর্বদাই সত্য, সুতরাং লূপটি একটি অসীম লূপ।

দ্র ঃ while (i + + < 10) এবং while (+ + i < 10) এই দুটির পার্থক্য কি?

প্রথম ক্ষেত্রে i-এর বর্তমান মানের সঙ্গে 10-এর তুলনা হবে এবং তার পর i-মান 1-বাড়বে, কিন্তু দ্বিতীয় ক্ষেত্রে প্রথমেই i-এর 1 বাড়বে এবং সেই বৃদ্ধিপ্রাপ্ত i-এর মানের সঙ্গে 10-এর তুলনা করা হবে।

do–while লূপ ও তার ব্যবহার

এই লূপটি while-লূপেরই একটি পরিবর্তিত রূপ। while-লূপে শর্তটি পরীক্ষা করা হয় লূপের প্রথমেই, কিন্তু do-while লূপে শর্তটির পরীক্ষা হয় লূপের শেষে। এই লূপের সাধারণ কাঠামোটি হলো ঃ

লূপের কাঠামো

```
do
{
s1 ;
s2 ;
.........
..........
sn ;
}
while (শর্ত)
```

পূর্বের উদাহরণটি do-while ব্যবহার করেও করা যেতে পারে ঃ

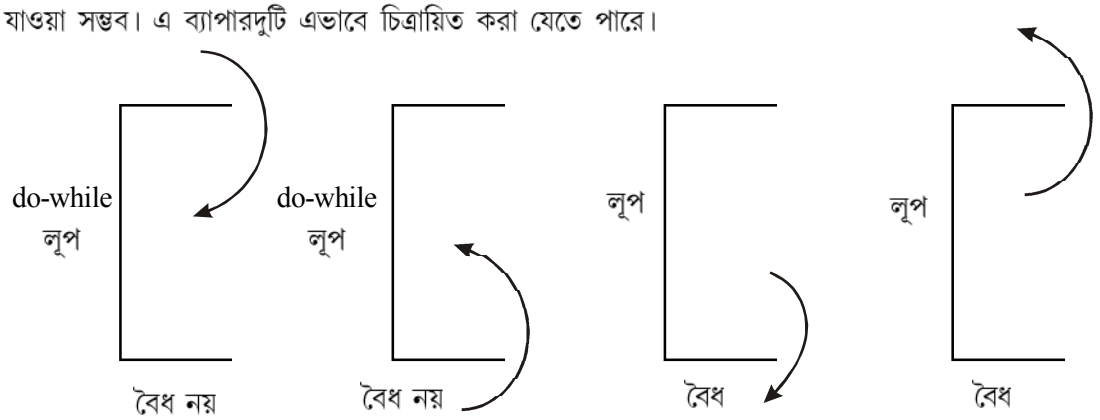
```
int n, sum ;
n = 51 ;
sum = 0 ;
do
{
sum = sum + pow (n, 3) ;
n = n + 2
}
while (n < = 99)
print f (" sum = % d ", sum) ;
```

যেহেতু লূপের শর্তটি লূপের শেষে পরীক্ষা করা হয়, সেই কারণে do-while লূপ অন্তত একবার আবর্তিত হবে।

do-while লূপ গঠনের ক্ষেত্রে কয়েকটি বিষয় মনে রাখা দরকার ঃ

১। বাইরে থেকে কোন নির্দেশের মাধ্যমে (যেমন goto ইত্যাদি) সরাসরি কোন do-while লূপের অভ্যন্তরে প্রবেশ করা যাবে না।

২। প্রয়োজন হলে do-while লূপের কাজ সম্পূর্ণ না করেই বিশেষ কোন শর্ত প্রয়োগ করে লূপের বাইরে যাওয়া সম্ভব। এ ব্যাপারদুটি এভাবে চিত্রায়িত করা যেতে পারে।



C–ভাষায় for–লূপ ও তার ব্যবহার

for-লূপ C–ভাষার ব্যবহৃত লূপ-কাঠামোগুলির মধ্যে সম্ভবতঃ সবচেয়ে বেশী ব্যবহৃত হয় এমন একটি লূপ। এই লূপে একই বিবৃতিতে লূপ-চলকের প্রাথমিক মান, অন্তিমমান এবং চলকের বৃদ্ধির (বা হ্রাসের) মান, সবই একসঙ্গে দেওয়া হয়। এর সাধারণরূপটি হলো

```
for (লূপ-চলকের প্রাথমিক মান প্রদান; অন্তিম মান বা অন্তিম মান নির্দেশক কোন শর্ত;
বৃদ্ধি (বা হ্রাস)-এর মান)
{
s1 ;
s2 ;
.....
.....
sn ;
}
```

উদাহরণ (১) ঃ

```
int sum = 0, x ;
for (x = 0, x <= 100; x++)
{
sum = sum + x ;
print ( " % d ", sum) ;
```

আউটপুটটি হবে প্রথম 100টি স্বাভাবিক সংখ্যার যোগফল (= 5050)

উদাহরণ (২) ঃ এই উদাহরণটি লক্ষ্য করা যাক ঃ

```
for (i = 0, j = 1; i + j <= 100; i+= 2, j++)
```

এখানে for-লূপের দুটি চলক, i এবং j —এদের প্রাথমিক মান প্রদানের সময় (,)-দ্বারা পৃথক করা হয়েছে।

প্রাথমিক মান এবং লূপের শর্তটির মধ্যে (;) চিহ্ন আছে, লূপের শর্ত এবং চলকগুলির মান-বৃদ্ধি (বা হ্রাস) নির্দেশগুলির মধ্যে (;) এবং মান-বৃদ্ধির (বা হ্রাসের) ক্ষেত্রে চলকগুলি (,)-দ্বারা পৃথকীকৃত।

উদাহরণ (৩) ঃ নীচের for-লূপটি বৈধ ঃ

```
for (i = (m + n) / 2 ; i <= 20 ; i = i / 2)
```

অর্থাৎ প্রাথমিক মান এবং মান-পরিবর্তনের নির্দেশ দুটিতেই রাশিমালার প্রয়োগ করা হয়েছে

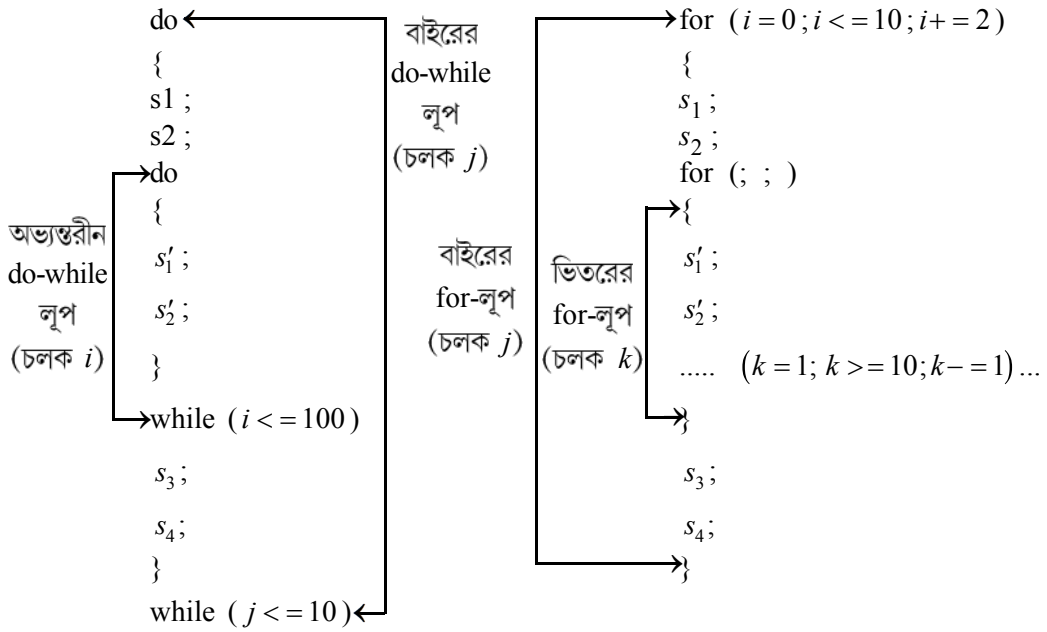
উদাহরণ (৪) ঃ for-লূপে প্রাথমিক মান এবং/অথবা মান-পরিবর্তনের নির্দেশ লূপের মধ্যেই কিন্তু for ( )-এর বাইরে করা যেতে পারে।

```
.....
.....
n = 1  /* প্রাথমিক মান প্রদান * /
```

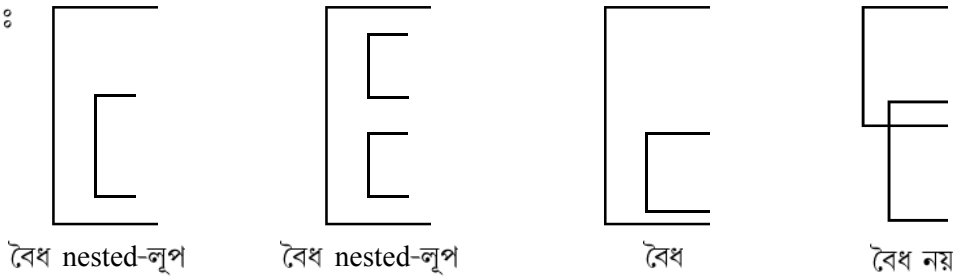
```
for ( ; n <= 10 ;)   / * for ( ) শুধুমাত্র শর্তটি লেখা হয়েছে * /
{
.....
.....
print f   (" product = % d", n * n) ;
n = n + 2 ;   / * লূপ চলকের মান-বৃদ্ধির নির্দেশ * /
}
```

Nested লূপ অর্থাৎ লূপের মধ্যে লূপ

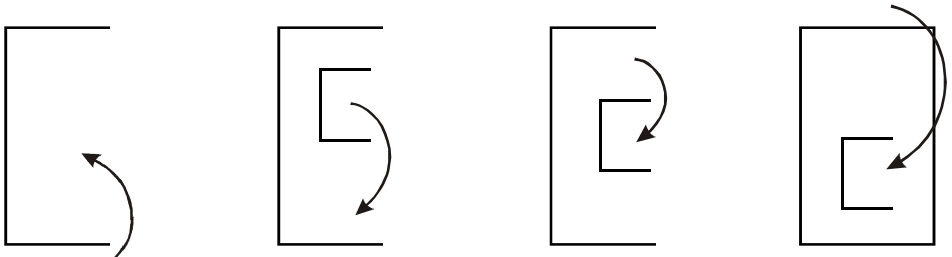
while, do-while, for-লূপ সমুহের কাঠামোতে লূপের অভ্যন্তরে লূপ থাকতে পারে। নীচের কাঠামোগুলি এই ধরনের nested লূপ ঃ



Nested-লূপের কাঠামোটি এমন হবে যাতে দুটি কাঠামো যেন পরস্পরছেদী না হয়। নীচের লূপ গুলি লক্ষ্য করো ঃ



বাইরে থেকে সরাসরি লূপের অভ্যন্তরে প্রবেশ করা যাবে না ঃ



C–ভাষাতে Array–চলনের ব্যবহার

একই ধরনের চলক সংখ্যার অনেক হলে তাদের পৃথক পৃথক নামে প্রকাশ না করে একই সাধারণ নামে প্রকাশ করা অনেক বেশী সুবিধাজনক, শুধুমাত্র ঐ সাধারণ নামের সাথে একটি পূর্ণসংখ্যা সূচক হিসাবে ব্যবহার করে তাদের পৃথকীকরণ করা হয়। যেমন ধরা যাক একই সমতলে দশটি বিন্দু আছে, যাদের স্থানাঙ্কগুলি দুভাবে লেখা যেতে পারে ঃ

$(a,b),(c,d),(e,f),(g,h),$ ..... ইত্যাদি

অথবা $(x_1,y_1),(x_2,y_2),(x_3,y_3),(x_4,y_4),.....,(x_{10},y_{10})$

নিঃসন্দেহে বিকল্প পদ্ধতিটি অনেক বেশী কার্যকরী এবং গ্রহণযোগ্য। প্রতিটি বিন্দুর ভূজ-কে x দ্বারা এবং কোটীকে y-দ্বারা চিহ্নিত করা হয়েছে। অর্থাৎ x হচ্ছে ভূজ-গুলির সাধারণ নাম, এবং y হচ্ছে কোটীর সাধারণ নাম। একটি বিন্দু থেকে আর একটি বিন্দুক আলাদা করার জন্য তাদের ভূজ এবং কোটীর সাধারণ নামের সঙ্গে একটি করে সাংখিক সূচক ব্যবহার করা হয়েছে, যেমন প্রথম বিন্দুর ভূজ x_1, দ্বিতীয় বিন্দুর ভূজ x_2 ইত্যাদি। এরকম ক্ষেত্রে আমরা x বা y-কে array-চলক বা subscripted-চলক বলে থাকি। এখানে এই array-চলকের একটি মাত্র সূচকই যথেষ্ট, যেমন $x_1,x_2,.....x_{10}$ ইত্যাদি, অর্থাৎ সূচকগুলি হলো 1, 2, 3, 10। এই ধরণের array-চলককে একমাত্রিক (one dimensional) array-চলক (অথবা List) বলা হয়। আবার একটি দ্বিমাত্রিক টেবল্-এর কথা ধরা যাক্ (যেমন ম্যাট্রিক্স) ঃ

$$\begin{matrix} & \downarrow \text{ কলাম (column)} \\ \text{সারি (row)}\rightarrow & \end{matrix} \begin{pmatrix} 10 & -3 & 5 & 7 \\ 2 & 0 & 9 & 5 \\ 7 & 11 & -5 & 4 \end{pmatrix}$$

এই দ্বিমাত্রিক (two-dimensional) টেবল্-এর সাধারণ রূপটি হলো

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & .....a_{1n} \\ a_{21} & a_{22} & a_{23} & .....a_{2n} \\ ..... & ..... & ..... & \\ a_{m1} & a_{m2} & a_{m3} & .....a_{mn} \end{pmatrix}$$

অর্থাৎ এই টেব্‌ল-এ মোট m সংখ্যক সারি (row) আছে এবং n-সংখ্যক কলাম্ (column)-আছে, মোট (m×n) সংখ্যক সংখ্যা আছে। কোন একটি সংখ্যার অবস্থান চিহ্নিত করতে হলে তার সারির নম্বর এবং কলাম্ নম্বর দুইই উল্লেখ করতে হবে। যেমন উপরের উদাহরণে চিহ্নিত সংখ্যাটি হলো a_{23} যার মান = 9 প্রতিটি সংখ্যাকেই একটি সাধারণ নাম *a*-দ্বারা চিহ্নিত করা হয়েছে, এবং কোন বিশেষ সদস্যকে চিহ্নিত করার জন্য দুটি করে সূচকের উল্লেখ প্রয়োজন।

পরিগণকীয় ভাষাতেও এই ধরনের array-চলকের ব্যবহার প্রায়শই হয়ে থাকে—C–ভাষাও তার ব্যাতিক্রম নয়। মনে রাখতে হবে এই array-চলকের মান শুধুমাত্র সাংখ্যিক হবে তা নয়, বর্ণ-মানও হতে পারে। C–ভাষাতে একমাত্রিক array-চলক লেখার পদ্ধতিটি হলো ঃ

(১) int mark [20] ;

এই নির্দেশে mark-একটি একমাত্রিক array-চলকের সাধারণ নাম বোঝানো হয়েছে যার অধীনে mark [0], mark [1], mark [2], ..., mark [19] এই মোট 20-টি চলক আছে—প্রতিটি চলকই পূর্ণসাংখিক। পরিগণক এই নির্দেশটি পেয়ে তার স্মৃতিকোষের 20-টি স্থান নির্দ্দিষ্ট করে রাখবে, mark [0], mark [1], ..., mark [19]-এই কুড়িটি পূর্ণ সাংখ্যিক চলকের জন্য, প্রতিটি স্থান হবে দুই-বাইটির যেহেতু চলকগুলি পূর্ণসাংখ্যিক। এই array-চলক mark-এর অন্তর্গত চলকগুলি mark [i]-র প্রাথমিক মান গুলিও একই সঙ্গে উল্লেখ করে দেওয়া যেতে পারে ঃ

int mark [20] = {67, 9, 19, 32, 87, ..., 10, 54, 67, 3, 32}, অর্থাৎ mark হলো একটা একমাত্রিক পূর্ণসাংখ্যিক array-চলক, যার অধীনে 20-টি পূর্ণসাংখ্যিক চলক আছে যাদের প্রাথমিক মানগুলি হলো ঃ

mark [0] =67, mark [1] =9, mark [2] = 19, ..., mark [17] = 67, mark [18] = 3, mark [19] = 32

লক্ষ্যণীয় যে mark [0] = 67 = mark [17], এবং এরকম একাধিক চলকের মান সমান হতেও পারে।

(২) float number [5] = {–3·85, 2·29, 0·42, 1·75}; এখানে number হলো array-চলকটির সাধারণ নাম, এর অন্তর্গত floating-চলকগুলি হলো number [0], number [1], ..., number [4] এবং এদের প্রাথমিক মানগুলি হলো

number [0] = –3·85
number [1] = 2·29
number [2] = 0·42
number [3] = 1·75
number [4] = 0·0 (উল্লেখিত নির্দেশে শুধুমাত্র 4-টি মানই দেওয়া আছে এক্ষেত্রে পরবর্তী চলকগুলি = 0 ধরা হয়)

(৩) char name [4] = {'O', 'P', 'E', 'N'} ;

এখানে name-একটি একক বর্ণ array-চলকের সাধারণ নাম। যার অন্তর্ভূক্ত বর্ণচলকগুলি হলো ঃ

name [0] = O, name [1] = P, name [2] = E, name [3] = N

এখানে উল্লেখকরা যেতে পারে গুরুবন্ধনীর অন্তর্গত সংখ্যাটি ঐ array-চলনের অন্তর্ভূক্ত চলকগুলির সর্বোচ্চ সংখ্যাটি নির্দেশ করে। এই নির্দেশক সংখ্যাটিকে স্পষ্ট উল্লেখ না ও করা যেতে পারে। যেমন—

int mark [ ] = {10, 14, 25, 89, 44}

float price [ ] = {3·35, 1·47, 13·4, 7·1, 2·2, –11·2, 10·1};

char name [ ] = {'S', 'U', 'B', 'H', 'A', 'S'}

এখানে প্রথম দুটি ক্ষেত্রে (অর্থাৎ সাংখ্যিক চলকের ক্ষেত্রে) array চলকদুটির সাইজ যথাক্রমে 5 এবং 7 ধরা হবে, অর্থাৎ যতগুলি মান ডানদিকে { }-এর মধ্যে অন্তর্গত আছে তার সমান। তবে তৃতীয় ক্ষেত্রে চলকটির সাইজ 6 + 1 = 7 ধরা হবে, অর্থাৎ যতগুলি সদস্য আছে তার থেকে 1 বেশী, অর্থাৎ এক্ষেত্রে

name [0] = S, name [1] = U, ..., name [5] = S এবং সবশেষের name [6] = \0—এই চিহ্নটিকে Null-চিহ্ন বলা হয় এবং এই চিহ্নটি তন্ত্রীটির শেষ হওয়াকে নির্দেশ করে। এই array-চলকটিকে এভাবেও লেখা যায় ঃ

char name [ ] = {SUBHAS} ;

Array চলক—পড়া এবং লেখা ঃ

একটি array-চলকের অন্তর্ভূক্ত চলকগুলির মান কিভাবে পরিগণকে ইন্‌পুট হিসাবে সন্নিবিষ্ট করা যায় এবং ঐ মানগুলিকে কিভাবে আউটপুটের মাধ্যমে লেখা যায়, সেগুলি এখানে আলোচনা করা হবে।

উপরের উদাহরণগুলিতে প্রাথমিক মানগুলি { }-র মধ্যে রেখে চলকগুলির মানগুলি পরিগণকের স্মৃতিতে রেখে দেওয়া যায়। তবে এই পদ্ধতিতে একটি অসুবিধা আছে—সেটি হলো নির্দেশাবলীটি শুধুমাত্র ঐ মানগুলির জন্যই প্রযোজ্য। মানের বিভিন্ন সেটের জন্য বারবার নির্দেশাবলীতে পরিবর্তন করতে হবে। এর পরিবর্তে scanf প্রক্রিয়ার সাহায্য নেওয়া যেতে পারে।

মনে করা যাক্ 15 জন ছাত্রের কোন বিষয়ের নম্বর পরিগণকের স্মৃতিতে সন্নিবিষ্ট করতে হবে। এর জন্য নীচের নির্দেশগুলি অনুসরণ করা যেতে পারে ঃ

```
# include < stdio·h >
  main ( )
  {
  int i, mark [15] ;
  for (i = 0; i < = 14 ; i + +)
  {                              / * { টি না দিলেও হতো * /
```

```
scanf (" %d ", & mark [ i ]) ;
}                          /* } টি না দিলেও হতো */
}
```

ধরা যাক ঐ 15-টি mark-এর মানগুলি হলো ঃ {12, 37, 51, 7, 65, 54, 87, 29, 41, 52, 51, 65, 12, 91, 62}

নির্দেশটি কার্যকরী হবার পূর্বে পরিগণক এই মানগুলি জানতে চাইবে (কিভাবে?—আগে আলোচনা করা হয়েছে), তখন এই মানগুলি পরিগণককে ইন্‌পুট-হিসাবে দিতে হবে ঃ 12 ↵

37 ↵

.....

62 ↵

এই মানগুলি কীভাবে আউটপুট হিসাবে লেখা যাবে?

এর জন্য print f ( )—নির্দেশের ব্যবহারই যথাযথ হবে ঃ

```
# include < stdio·h >
main ( )
{
int i, mark [15] ;                  পূর্ণসাংখ্যিক i-এর জন্য
for (i = 0; i < 15; i++)        /*  i < 15 এবং
{                                   i <= 14 একই অর্থবহ */
print f (" % d", mark [ i ] ;
}
}
```

আউটপুট টি একই লাইনে আসবে ঃ

12 37 51 7 65 54 87 29 41 52 51 65 12 91 62

(দুটি মানের মধ্যে সাধারণত দুটি করে ঘর ফাঁক থাকবে)

আউটপুটটিকে এক কলামেও (অর্থাৎ উল্লম্বভাবে) লেখা যেতে পারে তবে সেক্ষেত্রে print f ( ) টি হবে ঃ

```
print f (" % d \n ", mark [ i ] ) ;
```

দ্বিমাত্রিক Array–চলক ঃ

দ্বিমাত্রিক array-চলকের (টেব্‌ল্) কথা আগেই বলা হয়েছে। C–ভাষাতে এই ধরণের একটি array-চলককে কীভাবে পড়া বা লেখা যায় তার পদ্ধতিগুলি এখানে আলোচনা করা হবে।

উদাহরণস্বরূপ, একটি দ্বিমাত্রিক পরিসংখ্যা বিভাজন ছক নেওয়া হলো, যেখানে চলক দুটি হলো x (উচ্চতা সে.মি) এবং y (ওজন কে.জি)

মোট 253 জন ছাত্রর উচ্চতা (সে.মি) এবং ওজনের (কেজি)—একটি দ্বিমাত্রিক পরিসংখ্যা বিভাজন ছক নীচে দেওয়া হলো ঃ

y \ *x*	121-130	131-140	141-150	151-160
31-40	5	11	3	3
41-50	7	24	31	17
51-60	6	29	34	12
61-70	3	9	18	9
71-80	2	11	12	7

এই টেবিলে মোট 5টি সারি (row) এবং 4টি কলাম্ (column) আছে এই ছক্টির সাধারণ নাম দেওয়া যেতে পাতে student, অর্থাৎ student হলো একটি দ্বিমাত্রিক ছক (বা ম্যাট্রিক্স)—এতে মোট 20-টি অংশ বা উপাদান (element) আছে। এক একটি উপাদানকে চিহ্নিত করার জন্য দুটি সূচকের প্রয়োজন—ঐ উপাদানের সারি নম্বর (*i*) এবং কলাম্ নম্বর (*j*)। student (*i*, *j*) হলো *i* নম্বর সারিতে *j* নম্বর কলামে যে উপাদানটি আছে। C–ভাষাতে সারি এবং কলাম-এর নম্বর 0 থেকে শুরু হয়; অর্থাৎ উপরের এই ছকে *i* মান 0, 1, 2, 3, 4 এবং *j*-র মান 0, 1, 2, 3; এবং এই ছকে student (3, 2) = 18—যার অর্থ হলো 141 থেকে 150 সে.মি উচ্চতা এবং 61 থেকে 70 কে.জি ওজন বিশিষ্ট মোট ছাত্রের সংখ্যা হলো 18। C–ভাষায় এই ছক্টিকে লেখা হবে

```
int student [5] [4] ;
```

পরিগণকের স্মৃতিতে এই ছক্টি যে ভাবে সঞ্চিত হবে তার একটি চিত্ররূপ হলো ঃ

	কলাম [0]	কলাম [1]	কলাম [2]	কলাম [3]
সারি নং [0]	[0] [0] = 5	[0] [1] = 11	[0] [2] = 3	[0] [3] = 3
সারি নং [1]	[1] [0] = 7	[1] [1] =24	[1] [2] =31	[1] [3] =17
সারি নং [2]	[2] [0] = 6	[2] [1] = 29	[2] [2] = 34	[2] [3] = 12
সারি নং [3]	[3] [0] = 3	[3] [1] = 9	[3] [2] = 18	[3] [3] =9
সারি নং [4]	[4] [0] = 2	[4] [1] =11	[4] [2] = 12	[4] [3] =7

এই দ্বিমাত্রিক ছকটিকে ইন্‌পুট হিসাবে পরিগণকে সন্নিবেশ করানোর জন্য প্রয়োজনীয় নির্দেশটি হলো ঃ

```
int student [5] [4];
for (i = 0; i < 5; i++)
{
for (j = 0; i < 4; j++)
{
scan f (" % d ", &student [i] [j]) ;
}
}
```

i-এর লূপ

j-র লূপ

এখানে প্রথমে $i = 0$ দিয়ে শুরু হবে এবং j-লূপটি সম্পূর্ণ কাজ করবে $(j = 0, 1, 2, 3)$, এরপর $i = 1$ হবে এবং j-লূপটি সম্পূর্ণ হবে। এই পদ্ধতিতেই student-এর উপাদানগুলি ইন্‌পুট হিসাবে দেওয়া যাবে। অতএব মানগুলি টাইপ করার করার সময় সারি অনুযায়ী টাইপ করতে হবে ঃ

```
5 ↵
11 ↵      সারি [0]
3 ↵
3 ↵
7 ↵
24 ↵
31 ↵      সারি [1]
17 ↵
.....
.....
```

প্রশ্ন ১ ঃ এই মানগুলি কলাম্ অনুযায়ী ইনপুট দিতে হলে উপরের নির্দেশে কি কি পরিবর্তন করতে হবে?

এই ছকটিকে (ম্যাট্রিক্স) আউটপুট হিসেবে ম্যাট্রিক্স-আকারে লিখতে হলে printf ( )- প্রক্রিয়ায় অবশ্যই $\backslash n$ ব্যবহার করতে হবে। যেমন ঃ

```
int student [5] [4] ;
for (i = 0; i < 5; i++)
{
for (j = 0; j < 4; j++)
print f (" % d", student [i] [j]);
print f (" \n") ;
}
```

বিঃ দ্রঃ যেহেতু ম্যাট্রিক্সটির সব অংশগুলির মান সমান অঙ্কের নয়, কোনটি এক অঙ্কের কোনটি আবার দুই অঙ্কের, সেক্ষেত্রে প্রাপ্ত আউটপুটটি সামঞ্জস্যপূর্ণম্যাট্রিক্স আকারে হবে না; যেমন

5			1	1			3			3		
7			2	4			3	1			1	7
6			2	9			3	4			1	2
3			9			1	8			9		
2			1	1				1	2			7

এটিকে সঠিক আকারে লিখতে হলে প্রথম print f ( )-এ \ t ব্যবহার করতে হবে।

প্রশ্ন ঃ \ t ব্যবহার করে এই ছক্‌টিকে স্বাভাবিক সুষম ম্যাট্রিক্স আকারে লিখুন।

অপেক্ষক ঃ ব্যবহারকারী কর্তৃক সংজ্ঞায়িত অপেক্ষক

C–ভাষার স্মৃতিভাণ্ডারে যে সমস্ত অপেক্ষক স্থায়ীভাবে প্রতিষ্ঠিত, যেমন sqrt (x), sin (x), cos (x), exp (x), abs (x) ইত্যাদি, সেগুলির ব্যবহার ব্যাখ্যা করা হয়েছে। এর বাইরেও ব্যবহারকারী তাঁর নিজস্ব প্রয়োজনে কিছু কিছু অপেক্ষক তাঁর নিজের মতন করে সংজ্ঞায়িত করে নিতে পারেন। এই ধরনের অপেক্ষকগুলিকে 'Users' defined Functions' অর্থাৎ ব্যবহারকারী কর্তৃক সংজ্ঞায়িত অপেক্ষক বলা যেতে পারে। এই সংজ্ঞায়িত করায় পদ্ধতি বা নিয়মগুলি জেনে নেওয়া যাক্‌।

উদাহরণ (১) ঃ $f(x) = x^2 + xy^2$, যেখানে x ও y এবং অবশ্যই f floating-point সংখ্যা ঃ

```
float f (float x, float y)
{
return (x * x + x * y * y)
}
```

উদাহরণ (২) ঃ $\underline{|n}$-এর মান বের করতে হবে, n যেখানে ধনাত্মক পূর্ণসংখ্যা এর জন্য প্রয়োজনীয় অপেক্ষকটি হবে ঃ

```
long int fact (int n)
{
int i ;
long int prod = 1 ;
if (n > 1)
for (i = 2; i <= n ; n++)
prod * = i
return (prod) ; }
```

এখানে fact ( ) অপেক্ষকে n-এর মান বসালেই $\underline{|n}$-এর মান ফেরৎ পাওয়া যাবে।

উদাহরণ (৩) ঃ তিনটি পূর্ণসংখ্যা a, b, c আছে। একটি অপেক্ষক তৈরী কর যার আউটপুট হবে ঐ তিনটি পূর্ণসংখ্যার বৃহত্তম সংখ্যাটি। একটি অপেক্ষকের সাহায্যে এটি দুটি অংশে করা হচ্ছে—প্রথম অংশে একটি অপেক্ষকের সাহায্যে দুটি পূর্ণসংখ্যার বৃহত্তর সংখ্যাটি নির্ণয় করা হবে, পরে মূল নির্দেশাবলীতে সেই অপেক্ষকটির সাহায্যে বৃহত্তম সংখ্যাটি নির্ণয় করা হবে ঃ

```
# include < stdio · h >
------------------------------------------
            int max (int x, int y)
            {
প্রথম       int z ;
অংশ        z= (x > = y)? x : y;          /* এখানে z এর মান x এবং y-এর বৃহত্তরটির
            return (z) ;                   সঙ্গে সমান হবে * /
            }
------------------------------------------
            main ( )
            {
দ্বিতীয়      int a, b, c, d ;
অংশ        scan f (" % d % d % d", &a, &b, &c) ;
            d = max (a, b)
            print f (" \n maximum = % d", max (c, d)} ;
            }
------------------------------------------
```

বিঃ দ্রঃ d = max (a, b) এই নির্দেশে d = a ও b-র মধ্যে বৃহত্তরটি সঙ্গে সমান হবে (প্রথম অংশের max ( )-এর সাহায্যে)

এবং max (c, d)-টি a, b ও c-র মধ্যে বৃহত্তম সংখ্যাটি নির্দেশ করবে।

কয়েকটি C–নির্দেশাবলী

১। একটি ত্রিভুজের বাহু তিনটির দৈর্ঘ্য দেওয়া আছে। ত্রিভুজাকার ক্ষেত্রটির ক্ষেত্রফল নির্ণয় করতে হবে।

মনে করা যাক্ ত্রিভুজটির বাহু তিনটি হল a সেমি, b সেমি এবং c সেমি। ত্রিভুজাকার ক্ষেত্রটির ক্ষেত্রফল

$\Delta = \sqrt{s(s-a)(s-b)(s-c)}$ যেখানে s = অর্ধ-পরিসীমা $= (a+b+c)/2$

C–নির্দেশাবলী ঃ

```
# include < stdio · h >
# include < math · h >
main ( )
{
```

float a, b, c, s, area ;
scan f (" %f %f %f", & a, & b, & c) ;
$s = (a + b + c) / 2 \cdot 0$;

area = sqrt $\left(s*(s-a)*(s-b)*(s-c)\right)$

print f (" AREA = %f sq · cm", area) ;
getch ( ) ;
}

(২) একটি দ্বিঘাত সমীকরণের বীজগুলি নির্ণয় করতে হবে। মনে করা যাক্ $ax^2 + bx + c = 0$ প্রদত্ত দ্বিঘাত সমীকরণ। এর নিরূপক $= d = \sqrt{b^2 - 4ac}$, যদি $d \geq 0$ হয় তবে বীজ দুটি হবে বাস্তব এবং তাদের মান হবে $x_1 = (-b + \sqrt{d})/(2a)$ এবং $x_2 = \frac{-b - \sqrt{d}}{2a}$, $d < 0$ হলে, বীজ দুটি জটিল এবং তাদের বাস্তব অংশটি হবে $-\frac{b}{2a}$ এবং জটিল অংশ দুটি হবে $\pm i\frac{\sqrt{|d|}}{2a}$ বা $\pm i\frac{\sqrt{-d}}{2a}$

C–নির্দেশাবলী ঃ

include < stdio · h >
include < math · h >
main ( )
{
float a, b, c, d, x1, x2, rp, ip ;
scanf (" %f %f %f", &a, &b, &c) ;
$d = b*b - 4 \cdot 0*a*c$;

if $(d >= 0)$

{

x1 = (–b + sqrt (d))/ (2·0 * a)

x2 = (–b – sqrt (d))/ (2·0 * a)

print f (" Roots are real : $x_1 = \%f$ and $x_2 = \%f$ ", x_1, x_2);

golt_0 last ;
}
else

```
{
printf (" Roots are imaginary") ;
rp = –b/(2·0 * a);
ip = sqrt (–d)/(2·0 * a) ;
printf (" Real part %f Imaginary part %f", rp, it ) ;
}
getch ( ) ;
getch ( ) ;
last :
}
```

(৩) x-এর বিভিন্ন মানের জন্য অপেক্ষক $f(x)$-এর মান নির্ণয় ধরা যাক্ $f(x) = x^3 + 2\log_3(x+1)$ এবং x-এর প্রদত্ত মানগুলি হলো $x = 1{\cdot}0\ (0{\cdot}1)\ 2{\cdot}0$ (অর্থাৎ x-এর মান 1·0 থেকে শুরু হবে, 0·1 করে বাড়বে এবং অন্তিম মান হবে 2·0) প্রথমেই $f(x)$ অপেক্ষকের অন্তর্গত $\log_3$ ( )-কে $\log_e$ ( )-তে নিয়ে আসতে হবে। log-এর নিয়ম অনুসারে

$\log_3$ ( ) = $\log_e$ ( ) × $\log_3$ (e) (Base-পরিবর্তনের নিয়ম অনুসারে)

$$= \frac{\log_e( )}{\log_e(3)}$$

অতএব, $f(x) = x^3 + \dfrac{2\log(x+1)}{\log(3)}$

C–নির্দেশাবলী ঃ

```
# include < stdio · h >
# include < math · h >
main ( )
{
float x, f ;
for ( x = 1·0 ; x <= 2·0 ; x+ = 0·1 )
{
f = pow (x, 3) + 2·0 * log (x+1)/log (3);
print ("\nx = %–6·2f f(x)=% 10·4f", x, f ) ;
```

```
        getch ( );
        }
    }
```

প্রশ্ন ঃ এই নির্দেশগুলি 'Users' defined function—ব্যবহার করে লেখ।

(৪) দুটি পৃথক ধনাত্মক পূর্ণসংখ্যা m এবং n-এর গ. সা. গু এবং ল. সা. গু নির্ণয় কর।

দুটি ধনাত্মক পূর্ণসংখ্যার গ. সা. গু নির্ণয়ের ইউক্লীডিয় পদ্ধতিটির অ্যালগোরিদম্ আগেই দেওয়া হয়েছে। সেই অনুসারে গ. সা. গু-র C–নির্দেশাবলী লিখতে হবে। আবার আমরা জানি গ.সা.গু × ল.সা.গু = সংখ্যা দুটির গুণফল। সেই সূত্র ধরে ল.সা.গু নির্ণয় করতে হবে।

```
        # include < stdio · h >
        main ( )
        int m, n, r
        scanf (" %d  %d",  & m,  &n) ;
        if (m > n)
        {
        t = n
        n = m
        m =  t
        }
start  :  r =  n  % m ;
        if ( r  =  =  0)
        {
        printf (" The  HCF  is  =  %d",  m)
        goto  last ;
        }
        else
                {
                if (r  =  =  1)
                {print (" The  numbers  are  co-prime") ;
                goto  end ;}
                }
                else
                {  n  =  m
                   m  =  r
```

```
goto start
}
last : getch ( ) ;
end :
}
```

(৫) কয়েকটি বাস্তব সংখ্যা দেওয়া আছে। এই সংখ্যাগুলি ক্রমবর্ধমান/ক্রমহ্রাসমান পর্যায়ক্রমে সাজাতে হবে। ধরা যাক সংখ্যাগুলি হলো

13·2, 127·9, 7·1, 19·3, 77·0, 15·4, 19·3, 33·9, 67·2, 95·3 (মোট 10 টি)

এই সংখ্যাগুলিকে $a(i)$ দ্বারা চিহ্নিত করা হলো, i = 0, 1, 2, ..., 9. এবং এদের ক্রমবর্ধমান পর্যায়ক্রমে সাজাতে হবে।

প্রক্রিয়াটির পদক্ষেপ হবে এইরকম ঃ

(১) প্রথমে $a(0)$ এবং $a(1)$-এর মধ্যে তুলনা করা হল। যদি $a(0) \geq a(1)$ হয় তবে এরা পরস্পরের মধ্যে স্থান পরিবর্তন করবে। এই পরিবর্তনের পরে $a(1) > a(0)$ হবে। মনে রাখতে হবে স্থান পরিবর্তনের ফলে বর্তমান $a(0)$ মূল তালিকার $a(1)$-এর সঙ্গে সমান এবং বর্তমান $a(1)$ মূল তালিকার $a(0)$-এর সঙ্গে সমান। কিন্তু $a(0) < a(1)$ হলে স্থান পরিবর্তনের কোন প্রয়োজন নেই।

২। এই প্রক্রিয়াটি সম্প্রসারিত করে $a(1)$ সঙ্গে $a(2)$, $a(2)$-র সঙ্গে $a(3)$ এইভাবে অগ্রসর হতে হবে।

৩। তুলনাগুলি সম্পূর্ণ হলে $a(10)$ হবে সর্বোচ্চ সংখ্যাটি

৪। একই প্রক্রিয়া $a(0)$ থেকে $a(9)$ পর্যন্ত করলে দ্বিতীয় সর্বোচ্চ সংখ্যাটি হবে $a(9)$

৫। পরপর এইভাবে করে গেলে আমরা যে তালিকাটি পাবো সেটি ক্রমবর্ধমান তালিকা।

এর জন্য প্রয়োজনীয় C–নির্দেশগুলি হলো ঃ

```
# include < stdio · h >
   main ( )
   {int n = 10, i, j ;
   float a (10), t ;
   for (i = 0; i < n ; i + +)
   scanf (" %f ", & a (i));
   for (i = 0; i < = n – 2 ; i + +)
   {
```

a(i) এবং *a (i+1)*-এর স্থান পরিবর্তন করবার পদ্ধতিটি লক্ষ্য রাখুন—এখানে তৃতীয় আর একটি চলক t-র সাহায্য নেওয়া হয়েছে। অবশ্য *a(i) <a(j)* হলে এই স্থান পরিবর্তনের কোন প্রয়োজন নেই।

```
for (j = i + 1 ; j <= n –1, j++)
{
if (a [i] > a [j])
{
t = a[i];
a[i] = a[j];
a[j] = t;
}
for (i = 0, i < n; i++)
print f (" %f \n ", a (i))
}
```

(৬) নীচের নিয়মগুলো মেনে একটি টেলিফোন বিল তৈরী করতে হবে ঃ

গ্রাহকের নাম ঃ শ্রী বিমল ঘোষ

প্রথম 50টি কল ঃ ফ্রী

পরবর্তী 250টি কল ঃ 1-টাকা কল পিছু

বাকী কল ঃ Rs. 1·20 প্রতি কল

মাাসিক ভাড়া বাবদ ঃ Rs. 180-00

দেওয়া আছে ঃ প্রাথমিক কল-মিটারের পাঠ এবং অন্তিম কল-মিটারের পাঠ

C–নির্দেশগুলি হবে ঃ

```
/ * To prepare Telephone Bill * /
# include < stdio · h >
main ( )
{
int open_reading, closing_reading, rent = 180, unit; float charge;
char  name ;
name = " Bimal Ghosh";
unit = closing _ reading _ open _ reading ;
if (unit < = 50)
{
charge = 0·0 ;
```

```
else
{
if (unit < = 300)
charge = 1·0 * (unit – 50·0)
}
else
charge = 250 + 1·20 * (unit – 300) ;
print f (” Name : %s Total Charge = Rs %f”, name, charge + rent),
}
}
```

EMT–16 (Block–2) : Computer Programming and Numerical Practical

(1) সাংখ্যিক সমাকলন :

I-এর মান (4-দশমিক স্থান পর্যন্ত শুদ্ধ) সাংখ্যিক সমাকলনের সাহায্যে নির্ণয় কর ঃ

$$\int_{1\cdot2}^{3\cdot6} \frac{e^x}{1+x^2}\,dx$$

পদ্ধতি (১) ঃ ট্র্যাপিজয়ডালের সূত্রর সাহায্যে (n=12)

পদ্ধতি (২) ঃ সিম্পসনেব $\frac{1}{3}$-সূত্রের সাহায্যে (n=12)

সংকেত ঃ float f (float x) ;

(সিম্প্সনের $\frac{1}{3}$-সূত্র) {

return (exp (x)/ $(1+x*x)$) ;

scanf-এর সাহায্যে $a = 1\cdot2$, $b = 3\cdot6$/অথবা সরাসরি a, b-এর মান দিতে হবে।

int $n = 12$;

যেমন float $a = 1\cdot2$, $b = 3\cdot6$, h, y (20), $s1 = 0\cdot$, $s2 = 0\cdot$;

অথবা scanf (" %f %f", & a, & b) ;

$h = (b-a)/n$;

অপেক্ষকের মান নির্ণয়ের জন্য ঃ

$x = 1\cdot2$

for $(i = 0;\ i <= n,\ i++)$

{

y [i] = $f(x)$

$x = x + h$

}

for $(i = 1,\ i < n,\ i+= 2)$

$s1 = s1 + 4 * y[i]$;

for $(i = 2,\ i <= n-2,\ i+= 2)$

$s2 = s2 + 2 * y[i]$;

$s = (h/3\cdot0) * (y[0] + y[n] + s1 + s2)$;

(2) নীচের সমীকরণটির একটি বাস্তব বীজ নির্ণয় কর (চার দশমিক স্থান পর্যন্ত শুদ্ধ)

$$f(x) = e^x - 3kx + 1 = 0 \quad (\text{for } k = 1{\cdot}0\,(0{\cdot}1)\,1{\cdot}5)$$

পদ্ধতি (১) ঃ দ্বিবিভাজন পদ্ধতি (Bisection method)

পদ্ধতি (২) ঃ Regaula-Falsi

পদ্ধতি (৩) ঃ Newton-Raphson পদ্ধতি

সংকেত ঃ **Newton-Raphsan পদ্ধতি**

```
/ * Newton-Raphson Method * /
float f1 (float x, float k)
{
return (exp (x) – 3·0 * k * x + 1)
}
float f 2 (float x, float k)
{
return (exp (x) – 3·0 * k)
for (k = 1·0 ; k = 0·1 ; k + = 0·5) / * initial guess * /
{ x0 = 0·0
start :
h = f 1 (x0,k)/f 2 (x0,k)
x1 = x0 – h ;
if (fabs (x1 – x0) < = 0·00001)
goto end ;
else
{
x0 = x1
goto start
}
end : printf (" The root is % f, % 8·4 f, k, x 0)
}
```

(৩) সাধারণ অবকলন সমীকরণের সমাধান ঃ

$$\frac{dy}{dx} = \frac{x^2 + y^2}{x + y}$$, দেওয়া আছে $y(1{\cdot}0) = 1{\cdot}0$

নির্ণয় করতে হবে y (x)-এর মান যেখানে $x = 1{\cdot}0\,(0{\cdot}1)\,2{\cdot}0$

পদ্ধতি (১) ঃ Euler-এর সংশোধিত পদ্ধতি

(২) ঃ 4-মাত্রার Runge-Kutta পদ্ধতি

সংকেত ঃ Runge-Kutta পদ্ধতি

```
float f (float x, float y)
{
return ((x*x + y*y)/(x + y))
float x0 = 1·0, y0 = 1·0, h = 0·1, n = 10, k1, k2, k3, k4, k
x = x0; y = y0;
for (i = 0; i <= n, i++)
{
k1 = h*f(x,y);
k2 = h*f(x + h/2, y + k1/2);
k3 = h*f(x + h/2, y + k2/2);
k4 = h*f(x + h, y + k3);
k = (k1 + 2*+k2 + 2*k3 + k4)/6·0;
x = x + h
y = y + k
}
```

(৪) 'r'-নির্ণয় কর/অথবা $y = a + bx$ -এই সরলরেখাটি স্থাপিত কর Least Square পদ্ধতিতে ঃ

x	1·4	3·1	4·3	5·7	6·2	7·8	8·3	9·8
y	0·3	0·7	1·2	1·3	1·1	1·4	1·3	1·7

সংকেত ঃ এক্ষেত্রে প্রধানত যে মানগুলি নির্ণয় করতে হয় সেগুলি হল ঃ $\sum x_i$, $\sum x_i^2$, $\sum y_i$, $\sum x_i y_i$ ইত্যাদি; এর জন্য প্রয়োজনীয় C–নির্দেশগুলি হল ঃ

```
float s1 = 0; s2 = 0, s3 = 0 ইত্যাদি
for (i = 1; i <= n, i++)
scanf (" %f %f", & x[i], & y[i]),
for (i = 1, i <= n, i++)
{
s1 = s1 + x[i];          /* s1 = Σxi */
s2 = s2 + x[i]*x[i];     /* s2 = Σxi^2 */
s3 = s3 + y[i];          /* s3 = Σyi */
s4 = s4 + x[i]*y[i];     /* s4 = Σxiyi */
ইত্যাদি
```

$Y = a + bx$ —সরলরেখার ক্ষেত্রে

$$b = \left(n\sum x_i y_i - \sum x_i \sum y_i\right) \Big/ \left\{n\sum x_i^2 - \left(\sum x_i\right)^2\right\}$$

$$a = \frac{1}{n}\left(\sum y_i - b\sum x_i\right)$$

($n =$ প্রদত্ত বিন্দু সংখ্যা—উপরের উদাহরণে $n = 8$)

সংক্ষিপ্তসার

এই পাঠ, একক-পাঁচ এর পরবর্তি অংশ। এখানে C–ভাষায় কিভাবে নির্দেশাবলী লেখা হয় তার বর্ণনা উদাহরণসহ দেওয়া হয়েছে। পরিগণক ভাষার প্রধান বিবৃতি দুটি হল ইনপুট ও আউটপুট বিবৃতি। C–ভাষার বিভিন্ন ধরণের ইনপুট ও আউটপুট অপেক্ষকদ্বয়কে বিশদভাবে উদাহরণসহ বর্ণনা করা হয়েছে। তাছাড়া অনান্য বিবৃতি যেমন— if, goto, switch ইত্যাদিও উদাহরণসহ উপস্থাপন করা হয়েছে। যে কোন পরিগণক ভাষায় লূপ একটি গুরুত্বপূর্ণ বিবৃতি। ইহার যথেষ্ট ব্যবহার হয়ে থাকে। C–ভাষায় while, do-while এবং for এই তিনটি লূপ প্রধানত ব্যবহার করা হয়। এই তিনটি লূপ উদাহরণসহ আলোচনা করা হয়েছে। C–ভাষায় কিভাবে এক মাত্রিক ও দ্বিমাত্রিক array ব্যবহার করা হয়, তার লেখা ও পড়ার প্রণালী ইত্যাদিও আলোচনা করা হয়েছে। শেষে অনেকগুলি সমস্যা সমাধানের জন্য C–ভাষার প্রয়োজনীয় নির্দেশাবলী দেওয়া হয়েছে।

তথ্যসূত্র ঃ

১। কম্পুটার অ্যাপলিকেশন, নিহার চৌধুরী, বিশ্বাস বুক স্টল,

২। ডায়নামিক মেমোরী কম্প্যুটার কোর্স, দেবেন্দ্র সিং সিন্‌হাস, ফ্যুশন বুক্স।

৩। কমপিউটার ফান্ডামেন্টালস্ হার্ডওয়্যার ও প্রোগ্রামিং, সামুয়েল মল্লিক, সিস্‌টেক পাবলিকেশন্স

৪। Programming in ANSI C, বালাগুরুস্বামী।