

তথ্য ও যোগাযোগ প্রযুক্তি (মৌখিক) একাদশ ও দ্বাদশ শ্রেণি পঞ্চম অধ্যায় প্রোগ্রামিং ভাষা

প্রোগ্রাম : কম্পিউটারের সাহায্যে কোনো সমস্যা সমাধান করার জন্য একটি কর্মসূচীর প্রয়োজন। সেই কর্মসূচীর নাম হচ্ছে প্রোগ্রাম। এক বা একাধিক নির্দেশের সমষ্টিকে প্রোগ্রাম বলে।

প্রোগ্রামিং : প্রোগ্রাম রচনার কৌশলকে বলা হয় প্রোগ্রামিং।

প্রোগ্রামিং ভাষা : যে ভাষা ব্যবহার করে প্রোগ্রাম লিখা হয়, তাকে প্রোগ্রামিং ভাষা বলে।

আদর্শ প্রোগ্রাম : যে প্রোগ্রামে কম্পিউটার প্রোগ্রামের যাবতীয় গুণাবলি বিদ্যমান থাকে, সে প্রোগ্রামকে আদর্শ প্রোগ্রাম বলে।

আদর্শ প্রোগ্রামের ৪টি বৈশিষ্ট্য নিম্নরূপ :

- ১। প্রোগ্রামটি নির্ভুল।
- ২। যতটা সম্ভব সরল ও সংক্ষিপ্ত।
- ৩। যুক্তিপূর্ণ বিন্যাস।
- ৪। সহজে প্রোগ্রামটি পরিবর্তন, পরিবর্ধন ও ভুল সংশোধন করা যায়।

• বিভিন্ন স্তরের প্রোগ্রামিং ভাষা :

১। প্রথম প্রজন্মের ভাষা : (১৯৪৫-১৯৪৯) :

কম্পিউটারের নিজস্ব ভাষা হচ্ছে মেশিন ভাষা। এটি কম্পিউটারের নিম্নস্তরের ভাষা। 0 এবং 1 দ্বারা এই ভাষা গঠিত বলে একে বাইনারী ভাষা বলে। 0 এবং 1 দিয়ে তৈরি ভাষাকে যান্ত্রিক ভাষা বলে। কম্পিউটার কেবলমাত্র যান্ত্রিক ভাষা বুঝতে পারে। তাই অন্য ভাষার লেখা প্রোগ্রামকে অনুবাদক প্রোগ্রামের মাধ্যমে যন্ত্র ভাষায় পরিণত করে নিতে হয় এবং পরে কার্বে পরিণত করতে হয়।

২। দ্বিতীয় প্রজন্মের ভাষা : (১৯৫০-১৯৫৯) :

দ্বিতীয় প্রজন্মের ভাষার নাম Assembly ভাষা। এই ভাষাকে সাংকেতিক ভাষা বলে। 0 এবং 1 এর পরিবর্তে সাংকেতিক চিহ্ন ব্যবহার করা হয় বলে এই ভাষাকে সাংকেতিক ভাষা বলে।

• **Assembly ভাষা এবং মেশিন ভাষার মধ্যে তুলনা :**

১। মেশিন ভাষা 0 এবং 1 দিয়ে তৈরী । Assembly ভাষা সাংকেতিক চিহ্ন দিয়ে তৈরী ।

২। মেশিন ভাষা বুঝার জন্য Computer এর অনুবাদকের দরকার হয় না। অপরদিকে Assembly ভাষা বুঝার জন্য Computer এর অনুবাদকের দরকার হয় ।

৩। **তৃতীয় প্রজন্মের ভাষা : (১৯৬০-১৯৭০) :**

তৃতীয় প্রজন্মের ভাষার নাম উচ্চস্তরের ভাষা। সাংকেতিক ভাষা এবং যন্ত্র ভাষায় প্রোগ্রাম লিখা একজন প্রোগ্রামারের পক্ষে রীতিমতো অসুবিধাজনক বিধায় উচ্চস্তরের ভাষায় প্রোগ্রাম লেখার প্রয়োজনীয়তা দেখা দেয়। উচ্চতর ভাষায় লেখা প্রোগ্রাম যন্ত্র নির্ভর নয়। মানুষের বোধগম্য বর্ণ, সংখ্যা, বাক্য ব্যবহার করে উচ্চতর ভাষায় প্রোগ্রাম তৈরী করা হয়। ফলে যেকোনো প্রোগ্রামারের পক্ষে এই ভাষায় লেখা প্রোগ্রাম বুঝতে কষ্ট হয় না। এই ভাষায় অতি সহজে এবং দ্রুত প্রোগ্রাম লিখা যায়। এইজন্য উচ্চতর ভাষা জনপ্রিয়তা লাভ করে।

- বিভিন্ন রকমের উচ্চস্তরের ভাষা : c,c++, c-sharp Visual basic, Php, Java, Script, Oracle, Python etc

৪। **চতুর্থ প্রজন্মের ভাষা : (১৯৭১-১৯৭৯):**

১। এটি সংক্ষেপে 4GL নামে পরিচিত।

২। এটি Non-procedural ভাষা।

৩। এই ভাষায় শুধু বলে দিতে হয় কী ফলাফল প্রয়োজন।

৪। কিভাবে কার্য সম্পাদন করতে হবে তা ব্যাখ্যা করার প্রয়োজন নাই।

৫। এই প্রজন্মের ভাষার জন্য প্রসেসিং ক্ষমতা বেশি দরকার হয়।

৫। **পঞ্চম প্রজন্মের ভাষা : (১৯৮০- বর্তমান)**

এই প্রজন্মে মানুষের ভাষাকে প্রোগ্রামের ভাষা হিসেবে ব্যবহার করা হবে। এই জন্য বিজ্ঞানীগণ অনবরত চেষ্টা চালিয়ে যাচ্ছেন। কথোপকথনের রীতিতে কম্পিউটার ও ব্যবহারকারীর মধ্যে দ্রুত পরস্পরিক সংযোগ সৃষ্টির জন্য এই ভাষা ব্যবহার হবে।

- **সাংকেতিক ভাষা ও উচ্চস্তরের ভাষার মধ্যে তুলনা :-**

১। সাংকেতিক ভাষা লিখা হয় বিভিন্ন সাংকেতিক চিহ্ন দিয়ে আর উচ্চস্তরের ভাষা লিখা হয়

মানুষের বোধগম্য হরফ , শব্দ ও বাক্য দিয়ে।

২। সাংকেতিক ভাষায় এক মেশিনের লেখা প্রোগ্রাম আরেক মেশিনে চালানো যায় না। অন্যদিকে উচ্চতর ভাষায় লেখা প্রোগ্রাম যেকোনো কম্পিউটারে চালানো যায়।

• নিম্নস্তরের ভাষা ও উচ্চস্তরের ভাষার মধ্যে তুলনা :-

- ১। নিম্নস্তরের ভাষা লিখা হয় 0 এবং 1 দিয়ে । আর উচ্চস্তরের ভাষা লিখা হয় মানুষের বোধগম্য হরফ , শব্দ ও বাক্য দিয়ে ।
- ২। নিম্নস্তরের ভাষায় লিখা প্রোগ্রাম বুঝার জন্য Computer এর অনুবাদকের দরকার হয় না কিন্তু উচ্চস্তরের ভাষায় লিখা প্রোগ্রাম বুঝার জন্য Computer এর অনুবাদকের প্রয়োজন হয় ।

• অনুবাদক প্রোগ্রাম :

যে প্রোগ্রামের মাধ্যমে অন্য ভাষায় লিখিত প্রোগ্রামকে যন্ত্র ভাষায় পরিনত করা হয় তাকে অনুবাদক প্রোগ্রাম বলে । উৎস প্রোগ্রামকে বস্তু প্রোগ্রামে পরিনত করতে যে সফটওয়্যার এর প্রয়োজন তাকে অনুবাদক প্রোগ্রাম বলে ।

• উৎস প্রোগ্রাম :

যন্ত্র ভাষা ছাড়া অন্য ভাষায় লিখিত প্রোগ্রামকে উৎস প্রোগ্রাম বলে ।

• বস্তু প্রোগ্রাম : আর যন্ত্র ভাষায় লিখিত প্রোগ্রামকে বস্তু প্রোগ্রাম বলে ।

• অনুবাদক প্রোগ্রাম ৩ প্রকার । যথা :

1. Assembler .
2. Compiler.
3. Interpreter.

Assembler : Assembly ভাষায় লেখা প্রোগ্রামকে যন্ত্র ভাষায় রূপান্তরের জন্য এটি ব্যবহার করা হয় । বিভিন্ন Computer এর Assembler বিভিন্ন প্রকার । Computer Company কম্পিউটার বিক্রি করার সময় এই প্রোগ্রামটি ক্রেতাকে দিয়ে থাকে । Assembler টি সাধারণত rom স্মৃতিতে রাখা হয় । যার ফলে প্রোগ্রামটি মুছে যাওয়ার সম্ভাবনা থাকে না । এছাড়া এটি Disk এ ও সঞ্চিত রাখা যায় । যা পরবর্তীতে কম্পিউটারে ব্যবহার করা যায় ।

Compiler : Compiler হলো এক ধরনের অনুবাদক প্রোগ্রাম । যা High Level ভাষায়

লিখিত প্রোগ্রামকে যন্ত্রভাষায় রূপান্তর করে । Compiler দুই ধাপে অনুবাদকের কাজ সম্পাদন করে ।

ধাপ-১ : প্রতিটি লাইন পড়ে । উৎস প্রোগ্রামে যদি কোনো ভুল থাকে তা সংশোধন করার জন্য ব্যবহারকারীকে Error message দেয় ।

ধাপ-২ : Object প্রোগ্রামকে নির্বাহ করানো হয় । ফলাফল প্রদর্শনের জন্য প্রোগ্রাম একবার Compile হয়ে গেলে পরবর্তীতে আর Compile করার প্রয়োজন হয় না ।

Assembler ও Compiler অনুবাদ প্রোগ্রামের মধ্যে তুলনা :-

- ১। Assembler ধীরগতিসম্পন্ন অনুবাদক প্রোগ্রাম । আর Compiler দ্রুতগতি সম্পন্ন অনুবাদক প্রোগ্রাম ।
- ২। Assembler এ প্রোগ্রাম নির্বাহে সময় বেশি প্রয়োজন হয় । Compiler এ প্রোগ্রাম নির্বাহে সময় কম লাগে ।

Interpreter : Interpreter এক ধরনের অনুবাদক প্রোগ্রাম যা high level ভাষায় লিখিত প্রোগ্রামকে মেশিন ভাষায় রূপান্তর করে । Interpreter প্রথমে প্রোগ্রামের একটি লাইন পড়ে নেয় । যদি ব্যাকরণগত ভুল না থাকে তবে ঐ নির্দেশটি যন্ত্রভাষায় পরিনত হয় এবং পরে তা কার্যে পরিনত করে । তারপর পরবর্তী নির্দেশ পড়া শুরু করে ।

Compiler ও Interpreter এর মধ্যে তুলনা :-

- ১। Compiler একসাথে সকল নির্দেশকে যন্ত্রভাষায় পরিনত করতে পারে । Interpreter সম্পূর্ণ প্রোগ্রামকে একসাথে অনুবাদ করতে পারে না ।
- ২। Compiler ব্যবহার করলে প্রোগ্রাম একবার Compile হয়ে গেলে আর কম্পাইল করতে হয় না । Interpreter ব্যবহার করলে প্রোগ্রামের প্রতিটি নির্দেশের পূর্বে অনুবাদ করতে হয় ।

প্রোগ্রামের অংশ ৩টি । যথা :-

- ১। Input
- ২। procesing
- ৩। Output

প্রোগ্রামের তৈরীর ধাপ সমূহ :

- ১। সমস্যা নির্দিষ্টকরণ
- ২। সমস্যা বিশ্লেষণ
- ৩। প্রোগ্রাম ডিজাইন
- ৪। প্রোগ্রাম উন্নয়ন
- ৫। প্রোগ্রাম বাস্তবায়ন
- ৬। ডকুমেন্টেশন
- ৭। রক্ষণাবেক্ষণ

De-bugging : প্রোগ্রামের ভুল ত্রুটি খুঁজে বের করে তা সংশোধন করাকে De-bugging বলে।

প্রোগ্রামের ভুল তিন ধরনের। যথা :

- ১। Syntax error (ব্যাকরণগত ভুল)
- ২। Logical error (যুক্তিগত ভুল)
- ৩। Runtime error (নির্বাহজনিত ভুল)

Syntax error : যে ভাষায় প্রোগ্রাম লিখা হবে সে ভাষার নিজস্ব কতগুলো নিয়ম থাকে। নিয়ম বহির্ভূত কোনো কোডিং হয়ে থাকলে তাকে ব্যাকরণগত ভুল (Syntax error) হিসেবে বিবেচনা করা হয়। ব্যাকরণগত ভুল থাকলে কম্পিউটার Error message দেয় এবং প্রোগ্রামের কোন লাইনে ভুল আছে তাও জানিয়ে দেয়। ভুল সংশোধন করার পর তা মেশিন ভাষায় রূপান্তর করবে এবং পরে তা কার্যে পরিনত করবে।

Logical error: প্রোগ্রামে কোনো Logic লিখতে ভুল হলে ফলাফল আসবে কিন্তু তা সঠিক হবে না। এই ধরনের ভুলকে Logical error বলা হয়। এ ধরনের ভুল হলে Computer কোনো Error message দিবে না। এজন্য এই ভুল সংশোধন করা খুবই কঠিন।

যেমন : $x > y$ এর স্থলে $x < y$ লিখলে কম্পিউটার কোনো Error message দিবে না। কিন্তু ফলাফল প্রদর্শিত হবে।

রান টাইম ভুল : প্রোগ্রাম নির্বাহের সময় ইনপুট দিতে ভুল হলে ফলাফল ভুল আসবে। এ ধরনের ভুলকে রান টাইম error বলে।

যেমন :- পূর্ণসংখ্যার স্থলে দশমিক সংখ্যা ইনপুট দেওয়া।

Psudo Code : Pseudo একটি গ্রীক শব্দ। যার অর্থ ছদ্ম নাম বা Nick name বা যা সত্য নয়। প্রোগ্রাম রচনার সময় অনেকেই প্রোগ্রামের Pseudo code প্রণয়ন করে থাকেন। Pseudo code নির্দিষ্ট কোনো ভাষার উপর নির্ভরশীল নয়। এই পদ্ধতিতে একটি প্রোগ্রামকে এমন ভাবে উপস্থাপন করা হয় যেনো তা সকলেই বুঝতে পারে। Pseudo code কে Algorithm এর বিকল্প হিসেবে বিবেচনা করা হয়। চূড়ান্ত প্রোগ্রাম তৈরীর আগে Pseudo code তৈরী করা হয়।

Algorithm : Algorithm শব্দের অর্থ ধাপে ধাপে সমস্যার সমাধান। কোনো একটি নির্দিষ্ট সমস্যা সমাধানের জন্য যুক্তিসঙ্গত ও ধাপে ধাপে সমাধান করার পদ্ধতিকে Algorithm বলে।

Algorithm এর ৪টি শর্ত নিম্নরূপ :

- ১। Algorithm টি সহজবোধ্য হতে হবে।
- ২। প্রত্যেকটি ধাপ স্পষ্ট হতে হবে।
- ৩। সসীম সংখ্যক ধাপে সমস্যার সমাধান করতে হবে।

৪। ব্যাপক ভাবে প্রয়োগ উপযোগী হতে হবে।

Flow Chart : Flow Chart শব্দের অর্থ প্রবাহ চিত্র। Algorithm কে চিত্রের মাধ্যমে উপস্থাপন করাকে বলা হয় Flow Chart। Flow Chart এর মাধ্যমে কোনো সমস্যা সমাধানের জন্য কার্যক্রম সমূহ কিভাবে সম্পাদিত হবে তা ধারাবাহিক ভাবে চিত্রের মাধ্যমে উপস্থাপন করে দেখানো হয়।

Flow Chart দুই প্রকার যথা :-

১। System Flow Chart

২। Program Flow Chart

ফ্লোচার্টে ব্যবহৃত প্রতীক বা চিহ্ন সমূহের ব্যাখ্যা :-

১। টার্মিনাল প্রতীক (Terminal symbol) :-



এটি একটি ডিম্বাকৃতির প্রতীক। এটি প্রোগ্রামের শুরু এবং শেষ নির্দেশ করে থাকে। একটি প্রোগ্রামে একাধিকবার টার্মিনাল প্রতীক ব্যবহৃত হতে পারে।

২। গ্রহন বা নির্গমন প্রতীক (Input & Output symbol) :-



সামান্তরিক আকৃতির প্রতীককে গ্রহন/নির্গমন প্রতীক বলে। এর সাহায্য প্রক্রিয়া করণের জন্য ইনপুট চাওয়া হয় এবং প্রক্রিয়া করণের শেষে ফলাফল প্রদর্শন করানো হয়।

৩। প্রক্রিয়া করণ বা প্রসেস প্রতীক (Process symbol) :-



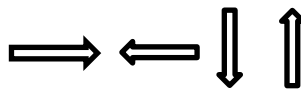
আয়তাকার আকৃতির প্রতীকই প্রক্রিয়াকরণ প্রতীক। প্রক্রিয়াকরণের কাজ কী রকম হবে তা নির্দেশ করার জন্য এই প্রতীক ব্যবহৃত।

৪। সিদ্ধান্ত প্রতীক (Decision symbol) :-





হীরক আকৃতির প্রতীকই সিদ্ধান্ত প্রতীক। এটি সর্বদা একটি প্রশ্ন করে থাকে যার দু'টি সম্ভাব্য উত্তর থাকে (হ্যাঁ বা না)। উত্তরের উপরের ভিত্তি করে ফ্লোচার্টকে দু'ভাগে বিভক্ত করা হয়।

৫। প্রবাহদিক প্রতীক (Flowline symbol) :-



তীরচিহ্ন বিশিষ্ট সরলরেখা প্রবাহদিক প্রতীক হিসেবে পরিচিত। তীর চিহ্নটি প্রোগ্রামের ব্যবহার পথ নির্দেশ করে। একটি অপারেশন চালিয়ে যাবার পর পরবর্তীতে কোথায় যেতে হবে তা নির্দেশ করে।

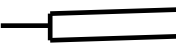
৬। সংযোগ প্রতীক (Corrector symbol) :-  এক পৃষ্ঠায়

 অপর পৃষ্ঠায়

বৃত্ত আকৃতির প্রতীকই সংযোগ প্রতীক হিসেবে ব্যবহৃত হয়। যখন একটি বড় ফ্লোচার্ট এক পৃষ্ঠায় ধরে না তখন সংযোগ প্রতীক ব্যবহার করে বাকী অংশ অপর পৃষ্ঠায় আঁকা হয়।

৭। সাবপ্রোগ্রাম প্রতীক (Subprogram symbol) :- 

মূল প্রোগ্রামের কোনো অংশ বা সাবপ্রোগ্রামকে এ প্রতীকের সাহায্যে চিহ্নিত করা যায়। একটি প্রোগ্রামে একাধিক সাবপ্রোগ্রাম থাকতে পারে।

৮। টীকা প্রতীক (Note symbol) :- 

ফ্লোচার্টের কোনো অংশ বর্ণনার জন্য অংশটির পার্শ্বে এ প্রতীক ব্যবহার করা হয়।

• অ্যালগরিদম ও ফ্লোচার্টের মধ্যে তুলনা :

১। অ্যালগরিদম বর্ণনা মূলক। ফ্লোচার্ট চিত্রমূলক।

২। অ্যালগরিদম প্রোগ্রাম রচনার ১ম ধাপ। ফ্লোচার্ট প্রোগ্রাম রচনার ২য় ধাপ।

• c- প্রোগ্রামের বৈশিষ্ট্য

১। c- প্রোগ্রামের শুরু হয় একটি main () ফাংশনের মাধ্যমে। প্রতিটি প্রোগ্রামের কাজ শুরু হয় এই ফাংশন থেকে।

২। ফাংশনের মধ্যে যে সকল statement থাকে সেগুলোকে দ্বিতীয় বন্ধনী { }- র ভেতর রাখতে হয়।

• Data type : c- প্রোগ্রামে অনেক ডেটা নিয়ে কাজ করা হয়, তবে প্রোগ্রাম চালনার সময় সব ডেটা মেমোরিতে সংরক্ষিত থাকে।

c- তে data চার প্রকার। যথা :-

1. Character data type

2. Integer data type (পূর্ণসংখ্যা)

3. Floating point data type. (ভাসমান বিন্দু-- যে সব সংখ্যার মধ্যে দশমিক বিন্দু থাকে)

4. Double data type.

• Character Data type :- এই ডেটার ব্যাপ্তি ০ হতে +255 পর্যন্ত। এক character লিখার জন্য মেমোরিতে 1 বাইট জায়গার প্রয়োজন হয়। 8 bit = 1 byte.

• Integer data type :- Memory তে Integer ডেটা লেখার জন্য 2 byte জায়গায় প্রয়োজন হয়। 16 bit = 2 byte

- **Floating point data type** :- Memory তে এই data লিখার জন্য 4 byte জায়গার প্রয়োজন হয়। 32 bit = 4 byte

- **Double data type** :- Memory তে এই data লেখার জন্য 4 byte জায়গায় প্রয়োজন হয়। 64 bit = 8 byte

Constant (ধ্রুবক) :- যে সমস্ত মান প্রোগ্রামে দিলে একই থাকে এবং প্রোগ্রাম নির্বাহের পরেও কোনো রূপ পরিবর্তন হয় না, সে সমস্ত মানকে ধ্রুবক বা Constant বলে।

Variable(চলক) :- প্রোগ্রাম নির্বাহের সময় যার মান পরিবর্তিত হয় তাকে চলক বলে।

আবার, একজন প্রোগ্রামার প্রোগ্রাম চলনাকালে কম্পিউটারের অস্থায়ী মেমোরিতে কিছু পরিমাণ জায়গা সংরক্ষিত হয়। অস্থায়ী মেমোরিতে ডেটা যে নির্দিষ্ট নামে সংরক্ষিত হয়ে থাকে, তাকে variable বলে।

Variable নাম করার নিয়মাবলী:-

1. চলকের নামের প্রথম অক্ষরটি অবশ্যই (A-Z, a-z) এর মধ্যে হতে হবে।
2. নামের দৈর্ঘ্য ৩১ অক্ষরের মধ্যে হতে হবে।

Variable দুই প্রকার। যথাঃ-

1. Local variable
2. Global variable

- **Local variable** :- যে সকল variable এর কার্যক্ষমতা কোনো নির্দিষ্ট ফাংশনের মধ্যে সীমাবদ্ধ থাকে তাকে Local variable বলে।
- **Global variable** :- যে সকল variable এর কার্যক্ষমতা কোনো নির্দিষ্ট ফাংশনের মধ্যে সীমাবদ্ধ নয় বরং প্রোগ্রামের যেকোনো অংশে ব্যবহার করা যায় তাকে Global variable বলে।

Global variable যেকোনো ফাংশনে ব্যবহার করা যায়। তবে এ variable কে main function এর পূর্বে ঘোষণা করা হয়।

- **Statement** :- যদি কোনো expression কিংবা ফাংশনের শেষে সেমিকোলন (;) ব্যবহার করা হয়, তখন c ভাষায় তাকে statement বলে।

Statement ২ প্রকার। যথা:-

১. Simple Statement
২. Compound Statement

- **Simple statement** :- যে statement এ একটি কিংবা একটি মাত্র ফাংশন থাকে তাকে Simple statement বলে। যেমন:-

a= 0;

a=a++;

print (“area”, area);

- **Compound statement**:- কতগুলো Simple statement কে যদি { } বন্ধনীর ভেতরে লেখা হয়, তখন তাকে Compound Statement বলে। যেমন:-


```
{  
a = 0;  
a= a++;  
print (“area”, area);  
}
```

- **Input/Output statement** :- কম্পিউটারের মাধ্যমে কোনো সমস্যা সমাধানের জন্য প্রোগ্রাম তৈরী করতে হয়। কম্পিউটারের সাহায্যে কোনো সমস্যা সমাধানের জন্য যে সব statement ব্যবহার করে ডেটা সরবরাহ করা হয় তাদেরকে ইনপুট statement বলে। আর কম্পিউটারের সাহায্যে কোনো সমস্যা সমাধানের পর যে ফলাফল পাওয়া যায় তা মনিটরে প্রদর্শন করে দেখানো জন্য যে সব statement ব্যবহার করা হয়, তাদেরকে Output statement বলে।
- **Getchar ()** :- এটি কী-বোর্ড হতে একটি character গ্রহন করার জন্য ব্যবহৃত হয়। Getchar () ফাংশনের ক্ষেত্রে keyboard থেকে যে character গ্রহন করা হয় তা মনিটরের পর্দায় দেখায় এবং Enter key-তে চাপ দেওয়ার জন্য অপেক্ষা করে। character-টি সবসময় single cotation এর ভেতর লিখতে হয়।
- **Getche ()** : keyboard হতে একটি character গ্রহণ করার জন্য এটি ব্যবহৃত হয় তবে Getchar () এর সাথে এর পার্থক্য আছে। Getchar () ফাংশনের ক্ষেত্রে keyboard থেকে যে character গ্রহন করে তারপর এটি মনিটরের পর্দায় দেখায় এবং Enter key-তে চাপ দেওয়ার জন্য অপেক্ষা করে। কিন্তু Getche () ফাংশনের ক্ষেত্রে keyboard থেকে যে character গ্রহন করা হয় তা মনিটরের পর্দায় দেখায় এবং Enter key-তে চাপ দেওয়ার জন্য অপেক্ষা করে না। character-টি সবসময় single cotation / ' ' এর ভেতর লিখতে হয়।
- **Getch ()** :- এটি কী-বোর্ড হতে একটি character গ্রহন করার জন্য ব্যবহৃত হয়। টাইপকৃত character-টি মনিটরে দেখা যায় না এবং Enter key তে চাপ দেওয়ার জন্য অপেক্ষা করে না।
- **Gets ()** :- এটি কী-বোর্ড হতে একটি string গ্রহন করার জন্য ব্যবহৃত হয়। একাধিক character এর সমন্বয়ে string গঠিত হয়। string সবসময় double cotation/ “ ” এর ভেতরে লিখতে হয়।
- **Putchar ()** :- এটি একটি character output statement। এর মাধ্যমে একটি character আউটপুট হিসেবে পাঠানো হয়। character-টি সবসময় single cotation এর ভেতর লিখতে হয়।
- **Puts ()** :- এটি একটি string output statement. এটি string চলকের মান Print করে
- **Scanf ()** :- এই ফাংশনের মাধ্যমে keyboard থেকে কোন মান, character, শব্দ/string ইত্যাদি Input নিয়ে কোনো variable-এ সংরক্ষণ করা যায় এবং পরে প্রসেস করে মনিটরে প্রদর্শন করা যায়।
- **Printf ()** :- এটি একটি formated output function। screen এ বা মনিটর এ কোনো কিছু দেখাতে হলে printf () এর বন্ধনীর ভেতর লিখতে হয় এবং লেখাটি double cotation এর ভেতর থাকে। অর্থাৎ, printf (“X”)
- **Keyword**:- প্রত্যেক ভাষার নিজস্ব কিছু শব্দ আছে। এ শব্দ গুলোকে বলে Keyword . প্রত্যেক Keyword এর সুনির্দিষ্ট কিছু অর্থ আছে এবং এ অর্থ অপরিবর্তনীয়। Keyword গুলো প্রোগ্রামে গুরুত্বপূর্ণ ভূমিকা পালন করে।

এ শব্দগুলোকে চলকের নাম হিসেবে ব্যবহার করা যায় না। C ভাষায় Keyword এর সংখ্যা অত্যন্ত কম এবং তা হলো ৩২টি।

যেমন :- Int, Char, Float, do, dowhile, For, else, elseif, endif. ইত্যাদি।

Format Specifier

কেন ব্যবহার করা হবে ?

- i) % c → একটি charcter Input-output এর জন্য
- ii) % d → Intger type data Input-Output এর জন্য
- iii) % o → অষ্টাল Int type data input-output এর জন্য
- iv) % x → হেক্সাডেসিমাল Int type data input-output এর জন্য
- v) % f → float type data input-output এর জন্য
- vi) % ld → Long Int type data input-output এর জন্য
- vii) % lF → Double type data input-output এর জন্য
- viii) % s String type data input-output এর জন্য
- ix) % n Out put নিচের লাইনে প্রদর্শনের জন্য অর্থাৎ, নতুন লাইন তৈরীর জন্য
- x) % t Out put 1 tab দূরে সরানোর জন্য [1 tab হচ্ছে এর space ৩ গুন]
- xi) \\ back slash প্রদর্শনের জন্য
- xii) \? প্রশ্নবোধক চিহ্ন প্রদর্শনের জন্য
- xiii) \b Out put বামে এক ঘর সরানোর জন্য
- xiv) \a alarm/ সংকেত প্রদর্শনের জন্য
- xv) \" \" double cotation (“ ”) চিহ্ন প্রদর্শনের জন্য
- xvi) \o null character প্রদর্শনের জন্য