সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

সি প্রোগ্রামারদের মাঝে এমন অনেকেই আছেন যারা প্রোগ্রামিংয়ের সম্পূর্ণ কোর্স শেষ করেছেন কিন্তু সি ল্যাঙ্গুয়েজের বেসিক ধারণা স্পষ্ট নয় বলে অনেক ক্ষেত্রেই বিভিন্ন সমস্যায় পড়েন। এ লেখায় মূলত সি ল্যাঙ্গুয়েজের বিভিন্ন প্রাথমিক উপাদান নিয়ে খুবই সংক্ষিপ্ত কিন্তু মৌলিক আলোচনা করা হয়েছে। আলোচনার বিষয়বস্তুর মাঝে আছে ডাটা টাইপ, ভেরিয়েবল, কিওয়ার্ড, অপারেটর ও এক্সপ্রেশন, স্টেটমেন্ট, লুপ, ফাংশন, অ্যারে এবং পয়েন্টার।

ডাটা টাইপ

একটি ভেরিয়েবল কী ধরনের ডাটা নিয়ে কাজ করবে, তা ঠিক করে দেয়া হচ্ছে ডাটা টাইপ। কানো ভেরিয়েবলের ডাটা হতে পারে কোনো পূর্ণসংখ্যা (যেমন ৪২, ৫৩ ইত্যাদি), কোনো ভগ্নাংশ (যেমন ৮.১৪ ইত্যাদি) অথবা কোনো অক্ষর বা character (যেমন ধ, ন, প ইত্যাদি)। সি-তে প্রধানত চার ধরনের ডাটা টাইপ থাকে। এগুলো হলো character (লিখতে হয় char), integer (লিখতে হয় int), float, double। মেমরির জায়গার ক্ষুদ্রতম একক হলো বিট, ৮ বিটে ১ বাইট, ১০২৪ বাইটে ১ কিলোবাইট, ১০২৪ কিলোবাইটে ১ মেগাবাইট, ১০২৪ মেগাবাইটে ১ গিগাবাইট ইত্যাদি। char টাইপ ডাটার জন্য ১ বাইট নির্ধারণ হয় এবং এ ধরনের ভেরিয়েবলে শুধু character রাখা যায়। int টাইপ ডাটার জন্য ২ বাইট নির্ধারণ হয় এবং এ ধরনের ভেরিয়েবলে শুধু পূর্ণসংখ্যা রাখা যায়। কিন্তু এ সংখ্যার মানের একটি লিমিট আছে। ১৬ বিটে কাজ করলে একেটটা ভেরিয়েবলে -৩২৭৬৮ থেকে +৩২৭৬৭ পর্যন্ত মান রাখা যায়। এটি বের করার একটি সূত্র হলো -2^n থেকে 2^n-1 পর্যন্ত। এখানে n হলো মোট বিটসংখ্যা। একটি ইন্টিজার টাইপ ভেরিয়েবল মোট ২ বাইট (১৬ বিট) জায়গা নেয়। তবে একটি সাধারণ ইন্টিজার ভেরিয়েবলের মানের সীমা 2^15 থেকে 2^15-1 পর্যন্ত। খেয়াল রাখতে হবে, ভেরিয়েবল যদিও ১৬ বিট জায়গা নিচ্ছে, কিন্তু সেটি ব্যবহার করছে ১৫ বিট এবং সবার বামদিকের ১টি বিট ব্যবহার করা হয় ভেরিয়েবলটির মান ধনাত্মক না ঋণাত্মক তা নির্ধারণ করার জন্য। float-এর জন্য ৪ বাইট নির্ধারণ হয় এবং এতে ভগ্নাংশ রাখা যায়। double-এও ভগ্নাংশ রাখা যায়, তবে তা ৮ বাইট জায়গা নেয়।

ভেরিয়েবল

কোনো তথ্য নিয়ে কাজ করার জন্য প্রথমে সেই তথ্যটিকে কমপিউটারের মেমরিতে রাখতে হয়। অর্থাৎ কমপিউটারের কাজ করার পদ্ধতিটি হলো প্রথমে মেমরিতে একটি সংখ্যা রাখা হলো (ধরা যাক, তথ্যগুলো কিছু সংখ্যা)। তারপর মেমরির আরেক জায়গায় আরেকটি সংখ্যা রাখা হলো। এবার সংখ্যা দুটির যোগফল বের করে সেই যোগফল মেমরির আরেক জায়গায় রাখা হলো। এবার যোগফলটি ইচ্ছে করলে প্রিন্ট করা যাবে, অর্থাৎ মনিটরে দেখানো যাবে। অথবা ইউজার চাইলে অন্য কোনো কাজও করতে পারেন। যেমন, অন্য কোনো পোর্টে সংখ্যাটি আউটপুট দেয়া। এখানে অন্য কোনো পোর্ট বলতে যেকোনো আউটপুট ডিভাইস বোঝানো হচ্ছে। যেমন প্রিন্টার, স্পিকার, ইন্টারনেট ইত্যাদি। কিন্তু এই যে সংখ্যাগুলো মেমরিতে রাখা হলো, এ কাজটি অতটা সহজ নয়। কমপিউটারের র‍্যাম হলো তার প্রধান মেমরি এবং এখানেই কমপিউটার সব ডাটা রাখে এবং তা ব্যবহার করে বিভিন্ন কাজ সম্পাদন করে। র‍্যামে অসংখ্য মেমরি সেল থাকে এবং এই সেলগুলোই মেমরির গঠনগত একক। প্রতিটি সেলের একটি নির্দিষ্ট অ্যাড্রেস থাকে। সংখ্যাগুলো এসব নির্দিষ্ট সেলে রাখা হয়। এখন এই যে সংখ্যাগুলো মেমরিতে রাখা হচ্ছে, এর মূল পদ্ধতি হলো প্রথমে প্রোগ্রামে একটি সংখ্যা ডিক্লেয়ার করে তার জন্য একটি নির্দিষ্ট অ্যাড্রেস নির্ধারণ করে দেয়া। কিন্তু এ কাজটি খুবই ঝামেলার, কারণ র‍্যামে লাখ লাখ মেমরি সেল থাকে। এ ঝামেলা দূর করার জন্য ব্যবহার করা হয় ভেরিয়েবল এবং এটি হাই লেভেল ল্যাঙ্গুয়েজের একটি বৈশিষ্ট্য। প্রোগ্রামে প্রয়োজন মতো এক বা একাধিক ভেরিয়েবল ডিক্লেয়ার করা যায় এবং ইউজার সেই ভেরিয়েবলের নাম নিজের ইচ্ছে মতো দিতে পারেন। কিন্তু এই নামকরণে কিছু নিয়ম মেনে চলতে হয়। যেমন :

- কোনো ভেরিয়েবলের প্রথম অক্ষর কখন কোনো সংখ্যা হতে পারবে না।
- ভেরিয়েবলের নামে underscore(_) এবং dollar sign($) ছাড়া অন্য কোনো special sign ব্যবহার করা যাবে না।
- ভেরিয়েবলের নামের মাঝে কোনো ফাঁকা জায়গা থাকতে পারবে না অর্থাৎ ভেরিয়েবলের নাম সবসময় একটি শব্দ হতে হবে।
- কোন keyword-এর নাম ভেরিয়েবলের নাম হিসেবে ব্যবহার করা যাবে না। যেমন : integer_type, auto_key, var1 ইত্যাদি ভেরিয়েবলের নাম হতে পারে।

কোনো ভেরিয়েবল ডিক্লেয়ার করার সাধারণ নিয়ম হলো data type_name;। যেমন int id_no; float mark; ইত্যাদি। তবে একই ধরনের অনেকগুলো ভেরিয়েবল ডিক্লেয়ার করতে হলে বারবার ডাটা টাইপ লিখতে হয় না। যেমন : int id, batch, code; ভেরিয়েবল ডিক্লেয়ার করার সময়ই তার মান নির্ধারণ করে দেয়া যায়, যেমন : int id=248;

অপারেটর ও এক্সপ্রেশন

কীবোর্ডের কিছু ক্যারেক্টারকে যেমন : +,-,*,/,>,<,= ইত্যাদি প্রোগ্রামে গাণিতিক, যৌক্তিক বা সম্পর্কসূচক কাজ করতে অথবা এ ধরনের কাজ নিয়ন্ত্রণ করতে ব্যবহার করা হয়। এ ধরনের ক্যারেক্টারকে অপারেটর। অপারেটর, কনস্ট্যান্ট এবং ভেরিয়েবলের সঠিকভাবে প্রোগ্রামে উপস্থাপনের মাধ্যমে এক্সপ্রেশন তৈরি করা হয়। যেমন : int a,b=10; a=b; এখানে প্রথম লাইনে দুটি ভেরিয়েবল a এবং b ডিক্লেয়ার করা হয়েছে, যেখানে b-এর মান ১০ নির্ধারণ করে দেয়া হয়েছে। দ্বিতীয় লাইনে b-এর মান a-এর জন্য নির্ধারণ করা হয়েছে। এখানে দ্বিতীয় লাইনটি একটি এক্সপ্রেশন।

অপারেটরকে কয়েক ভাগে ভাগ করা যায়। তবে যেগুলো সি-তে সবচেয়ে বেশি ব্যবহার হয়, সেগুলোই শুধু দেয়া হলো :

- অ্যারেথমেটিক অপারেটর : +, -, ×(গুণ), /(ভাগ), %(modulus বা ভাগশেষ)।
- রিলেশনাল অপারেটর : >, <, >=, <=, !=(সমান নয়), ==(সমান)
- লজিক্যাল অপারেটর : !(not), &&(and), ||(or)

কিওয়ার্ড

সি-তে কিছু সংরক্ষিত শব্দ আছে যেগুলোকে বলে কিওয়ার্ড। এসব শব্দ ব্যবহার করে কম্পাইলার কিছু নির্দিষ্ট কাজ সম্পাদন করেন। এসব সংরক্ষিত শব্দকে এদের নির্দিষ্ট কাজ ছাড়া অন্য কোনো কাজে (যেমন কোনো কিছুর নাম হিসেবে) ব্যবহার করা যায় না। ANSI-এর মান অনুযায়ী সি-তে ৩২টি কিওয়ার্ড আছে। যেমন : auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, sized, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while। এছাড়া টার্বো সি-এর কিছু নিজস্ব কিওয়ার্ড আছে। যেমন : asm, cdecl, far, huge, interrupt, near, pascal, _cs, _ds, _es, _ss। এসব কিওয়ার্ডের প্রত্যেকটিরই কিছু বিশেষ কাজ আছে। যেমন int কিওয়ার্ড দিয়ে কোন ভেরিয়েবলের ডাটা টাইপ নির্ধারণ করা যায়। এখানে উল্লেখ্য, ওপরের বর্ণিত সব কিওয়ার্ডই কিন্তু ছোট হাতের অক্ষরে লেখা। তাই প্রোগ্রামের মাঝে কোনো কিছুর নাম হিসেবে iNt বা inT ব্যবহার করলে কোনো এরর দেখাবে না। এজন্য সি-কে কেস সেনসিটিভ বলা হয়। আরও লক্ষণীয়, main শব্দটি কোনো কিওয়ার্ড না হলেও এটি এমন একটি শব্দ, যা প্রতিটি প্রোগ্রামে অবশ্যই থাকতে হবে। কারণ কম্পাইলার সবসময় main() ফাংশন থেকে কম্পাইল করা শুরু করে। এছাড়া main-কে কখনও কোনো কিছুর নাম হিসেবেও ব্যবহার করা যায় না। এটি একটি ব্যতিক্রম।

স্টেটমেন্ট

সি-তে কোনো লাইনের শেষ বোঝাতে সেমিকোলন (;) ব্যবহার করা হয়। একটি কমপ্লিট এক্সপ্রেশন যখন সেমিকোলন দিয়ে শেষ করা হয়, তখন সি-তে তাকে স্টেটমেন্ট বলে। স্টেটমেন্ট সাধারণত দুই ধরনের হয়ে থাকে, যেমন :

সিম্পল স্টেটমেন্ট : একটিমাত্র এক্সপ্রেশন বা ফাংশন নিয়ে যে স্টেটমেন্ট গঠিত হয় তা হলো সিম্পল স্টেটমেন্ট।

কম্পাউন্ড স্টেটমেন্ট/কোড ব্লক : একাধিক স্টেটমেন্টকে যখন দ্বিতীয় বন্ধনীর '{}' মধ্যে লেখা হয়, তখন তাকে কম্পাউন্ড স্টেটমেন্ট বা কোড ব্লক বলে। সিম্পল ও কম্পাউন্ড স্টেটমেন্টের মধ্যে বিশেষ কোনো পার্থক্য নেই, শুধু স্কোপ বা কার্যক্ষেত্রের ভিন্নতা দেখা যায়।

লুপ

সি-তে লুপ ব্যবহারের জন্য সাধারণত if else, for, while, do while ইত্যাদি ব্যবহার করা হয়।

if else স্টেটমেন্ট তিনভাবে ব্যবহার করা যায়। যেমন : শুধু if, if এবং else, else if চেইন। উদাহরণ হিসেবে তিনটি ছোট প্রোগ্রাম দেয়া হলো :

শুধু if

```
if(age>=18)
        printf("You are mature.");
if(age<=18)
        printf("You are immature.")
```

else if

```
If(age>=18)
        printf("You are mature.");
```

else

```
        printf("You are immature.")
```

else if চেইন

```
if(age>=50)
        printf("You are old.");
else if((age>=25)&&(age<50))
        printf("You are young");
else if((age>=18)&&(age<25))
        printf("You are mature");
else if((age>=10)&&(age<18))
        printf("You are a boy");
else if((age<10)&&(age>0))
        printf("You are a child");
else
        printf("You are not born!!");
```

লক্ষণীয়, এলস ইফ চেইনের শুরু হয় ইফ দিয়ে, ভেতরে থাকে এলস ইফ এবং শেষে থাকে শুধু এলস।

while লুপের শুরুতে কন্ডিশন থাকে, পরে কোড ব্লকের ভেতরে কোড থাকে। যেমন :

```
int x=1;
while(x!=100)
{
        printf("%d ",x);
        x++;
}
```

do while লুপের কাজ while লুপের মতোই, শুধু পার্থক্য হলো while লুপের শুরুতে কন্ডিশন চেক করে, তারপর কোড ব্লকে ঢোকে। আর do while লুপে প্রথমে একবার কোড ব্লক রান করার পর থেকে কন্ডিশন চেক করা হবে। যেমন :

```
printf("press any key to print and 'q' to
quit:\n")
char ch;
do
{
        ch=getch();
        printf("%c\n",ch);
}while(ch!='q')
```

ফর লুপের ব্যবহার সবচেয়ে বেশি হয়ে থাকে। এটি ব্যবহারের নিয়মও সহজ। যেমন :

```
int i=0;
for(i=0;i<10;i++)
{
        printf("%d\n",i);
}
```

ফাংশন

সি-তে কোড লেখার সময় সবাই main() এই অংশটি লেখে, যা লেখা দরকার। এটি একটি ফাংশন, যাকে মেইন ফাংশন বলা হয়। এটি ছাড়া যেমন প্রোগ্রাম রান করা সম্ভব নয়, তেমনি আরও অনেক ফাংশন আছে যেগুলো প্রোগ্রামের জটিলতা বহুগুণে কমিয়ে দেয়। এদের কাজ একই এবং বিভিন্ন সময় একই ধরনের কাজ করার জন্য এদের ব্যবহার করা হয়। সি-তে printf(), scanf(), clrscr() ইত্যাদি বিভিন্ন ফাংশন আছে। এসব ফাংশনের কাজ একই। যেমন printf()-এর কাজ কোনো কিছু প্রিন্ট অর্থাৎ মনিটরে দেখানো, scanf()-এর কাজ হলো ইউজারের কাছ থেকে কীবোর্ডের কোনো ইনপুট নেয়া, clrscr()-এর কাজ হলো স্ক্রিনে, যা কিছু আছে সব মুছে ফেলা ইত্যাদি। এ কাজগুলো আসলে এত সহজ নয়, যেমন ইনপুট নেয়ার জন্য সি-তে অনেক কোড লেখার প্রয়োজন। কিন্তু শুধু scanf() লিখলেই সহজে ইনপুট নেয়া যায়, কারণ এর জন্য প্রয়োজনীয় কোড আগে থেকে লিখে দেয়া হয়েছে। stdio.h নামের হেডার ফাইলে এই scanf() ফাংশনটি লেখা আছে। প্রোগ্রামে যখন scanf() লেখা হয়, তখন প্রোগ্রাম উক্ত হেডার ফাইল থেকে সংশ্লিষ্ট ফাংশনের কোডগুলো কম্পাইল করে নেয়। এভাবে ফাংশনের কাজই হলো সম্পূর্ণ প্রোগ্রামটিকে ধাপে ধাপে সম্পন্ন করা।

ফাংশন মূলত দুই ধরনের ঃ লাইব্রেরি ফাংশন এবং ইউজার ডিফাইন্ড ফাংশন। লাইব্রেরি ফাংশনের আরেক নাম হলো বিল্টইন ফাংশন। হেডার ফাইলে যেসব ফাংশন বর্ণিত থাকে, সেগুলো হলো লাইব্রেরি ফাংশন। এ ফাংশনগুলো আগে থেকেই লেখা আছে দেখে এরূপ নামকরণ। কম্পাইলার অনুযায়ী লাইব্রেরি ফাংশন নির্ধারিত হয়। তবে বেশিরভাগ ফাংশনই সব কম্পাইলারে অপরিবর্তিত থাকে। আবার ইন্টারনেটে অনেক এক্সটার্নাল লাইব্রেরি ফাংশনও পাওয়া যায়। এগুলো ব্যবহার করে প্রোগ্রাম আরও সহজে চালানো সম্ভব। আরেক ধরনের ফাংশনের নাম হলো ইউজার ডিফাইন্ড ফাংশন। এদের মাঝে আসলে তেমন কোনো পার্থক্য নেই। লাইব্রেরি ফাংশন হলো যেগুলো আগে থেকে বর্ণিত থাকে সেগুলো। আর ইউজার ডিফাইন্ড ফাংশন হলো ইউজার যে ফাংশনগুলো নিজের সুবিধার জন্য বানিয়ে নেয় সেগুলো। ইউজার ডিফাইন্ড ফাংশনের একটি উদাহরণ নিচে দেয়া হলো :

```
#include<stdio.h>
void func();
void main()
{
        clrscr();
func();
}
void func()
{
        printf(Òa user defined function is
created");
}
```

অ্যারে

অ্যারে হলো অনেকগুলো ভেরিয়েবল একসাথে ডিক্লেয়ার করার একটি পদ্ধতি। ধরা যাক, কোনো প্রোগ্রামে একইসাথে পাঁচটি ভেরিয়েবল ডিক্লেয়ার করার প্রয়োজন হলো। তাহলে ইউজার সাধারণ নিয়মে পাঁচটি ভেরিয়েবল ডিক্লেয়ার করতে পারেন। এজন্য পাঁচটি স্টেটমেন্ট লেখার প্রয়োজন হবে। কিন্তু অ্যারে ব্যবহার করে পাঁচটি ভেরিয়েবল একইসাথে অর্থাৎ একটি স্টেটমেন্ট দিয়েই ডিক্লেয়ার করা সম্ভব। মাত্র পাঁচটি ভেরিয়েবলের ক্ষেত্রে হয়তো এটি তেমন গুরুত্বপূর্ণ বিষয় নয়, কিন্তু বড় বড় প্রোগ্রামে একইসাথে যখন ১০০ বা ১০০০টি ভেরিয়েবল ডিক্লেয়ার করার প্রয়োজন পরে, তখন অ্যারে ব্যবহার করলে কোডিং অনেক সহজ হয়ে পড়ে। ভেরিয়েবল ডিক্লেয়ারের মতোই অ্যারে ডিক্লেয়ার করতে হয়। যেমন : int prime[10], valid[5]; ইত্যাদি।

পয়েন্টার

পয়েন্টার হলো একটি বিশেষ ভেরিয়েবল, কিন্তু এটি কোনো সাধারণ মান ধারণ করতে পারে না। এটি শুধু অপর ভেরিয়েবলের অ্যাড্রেস ধারণ করতে পারে। একটি প্রোগ্রামের জন্য অ্যাড্রেসটিই মূল বিষয়। প্রতিটি ভেরিয়েবলেরই একটি করে অ্যাড্রেস থাকে। প্রোগ্রাম ওই ভেরিয়েবলগুলোকে তাদের নামে নয়, বরং তাদের অ্যাড্রেস দিয়ে চেনে। ওই অ্যাড্রেসে কোনো কিছু পরিবর্তন করলে সংশ্লিষ্ট ভেরিয়েবলেও সেই পরিবর্তন দেখা যাবে। অর্থাৎ কোনো ভেরিয়েবলের যে অ্যাড্রেস আছে, সে অ্যাড্রেসের মানকে মুছে দেয়া হলে ভেরিয়েবলের মানও ডিলিট হয়ে যাবে। আবার কোনো অ্যাড্রেসে নতুন কোনো মান অ্যাসাইন করা হলে ওই অ্যাড্রেসের যে ভেরিয়েবল আছে তার মানও পরিবর্তন হয়ে যাবে।

যেকোনো ল্যাঙ্গুয়েজের ওপর দক্ষতা আনতে হলে সবার আগে তার বেসিক ধারণা সম্পর্কে পরিষ্কার হতে হবে কজ

ফিডব্যাক : wahid_cseaust@yahoo.com