

সি এর প্রথম পার্থ: ইনপুট হতে আউটপুট (পর্ব ৪)

আমরা আগের টিউটোরিয়ালে দেখলাম যে কি করে * দিয়ে পিরামিড বানাতে হয়। উক্ত টিউটোরিয়ালে যে সকল প্রোগ্রাম রয়েছে তাতে যদি printf ফাংশনের ভিতর কোন স্পেস ব্যবহার না করে নিউ লাইন কমান্ড (\n) ব্যবহার করি অর্থাৎ printf("*\n") আকারে ব্যবহার করি তাহলেও তা দিয়ে আউটপুটে সমকোণী ত্রিভুজের ন্যায় দেখানো সম্ভব। তখন প্রোগ্রামটি হবে নিম্নরূপ:

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
void main()
{
printf("*\n");
printf("***\n");
printf("*****\n");
printf("*****\n");
printf("*****\n");
printf("*****\n");
getch();
}
```

এই প্রোগ্রামের আউটপুট হবে নিম্নমত:

```
*
***
*****
*****
*****
```

যা দেখতে একটি সমকোণী ত্রিভুজের মত।

গত টিউটোরিয়ালের এই প্রোগ্রামটি আবার আলোচনায় আনার উদ্দেশ্য হল আজকে আমরা একই পিরামিডকে বা ত্রিভুজকে * ছাড়া অন্য কোন সিম্বল দিয়ে আঁকব। উদাহরণ স্বরূপ বলা যেতে পারে যে * এর পরিবর্তে a বসাব। তখন উপরের ত্রিভুজটি হবে নিম্নরূপ:

```
a
aaa
aaaaa
aaaaaaa
aaaaaaaa
```

আবার কেউ মনে করতে পারেন যে a ব্যবহার না করে x কে দিতে আঁকব। সেক্ষেত্রে ত্রিভুজটি দেখতে হবে নিম্ন আকারের।

```
x
xxx
xxxxx
xxxxxxx
xxxxxxxxx
```

এমনও হতে পারে যে কেউ হত চাইল যে সে 0, 1, 2, 3 এবং 4 অংকগুলি ব্যবহার করে নিম্নের ন্যায় ত্রিভুজ আঁকবে।

```
0
111
22222
3333333
444444444
```

এভাবে যদি চাওয়ামত সিম্বল ব্যবহার করে নিজের মত করে আঁকতে চাই (মানে একই প্রোগ্রাম বারবার রান করব, শুধু একটবার আমার টার্গেট সিম্বলকে দেখিয়ে দেব আর প্রোগ্রাম আমার দেখানো সিম্বল দিয়েই ত্রিভুজ আঁকবে। সেটি হতে পারে 0 দিয়ে, 1 দিয়ে, * দিয়ে বা অন্য যে কোন সিম্বল দিয়ে।) তাহলে স্থির কোন সিম্বল

ব্যবহার না করে কিবোর্ড হতে ইনপুট নিতে হবে। সেক্ষেত্রে কোন কেবোর্ডকে ইনপুট হতে নিয়ে আঁকা যাবে। কিন্তু কোন গাণিতিক কাজ করতে হলে, যেমন মিডটার্মের নম্বর ও ফাইনালের নম্বর যোগ করা, ডাটার ধরণ কেবোর্ড হলে চলবে না। সেক্ষেত্রে ডাটাকে অবশ্যই নম্বর হতে হবে। কাজেই বুঝা গেল যে ডাটা বিভিন্ন ধরণের হতে পারে। আবার ডাটাকে মেমরিতে কোন জায়গায় রেখে কাজ করতে হবে। মেমরির একই জায়গায় এক এক বার এক এক মান রেখে কাজ করার প্রয়োজন হতে পারে। যেমন মেমরির একটি জায়গা নির্ধারিত হলো মিডটার্মের নম্বরের মান রাখার জন্য। অন্য একটি জায়গা নেওয়া হলো ফাইনাল পরীক্ষার নম্বর রাখার জন্য। কোন কোর্স মেথ১০১ এর মিডটার্মের নম্বর মেমরির প্রথম অংশে ও এই কোর্সের ফাইনালের নম্বর মেমরির দ্বিতীয় অংশে রাখা হলো। তারপর মেমরির দুই নির্ধারিত স্থান হতে মান নিয়ে যোগ করে মেথ১০১ কোর্সে মোট কত পেল তা মনিটরে প্রকাশ করা হলো। এবার আর মেমরির মিডটার্ম ও ফাইনাল নম্বর রাখার স্থানটুকু মেথ১০১ কোর্সের জন্য ধরে রাখার প্রয়োজন নেই। এর জন্য প্রসেসিং এর কাজ শেষ। তাই পরের ধাপে একই স্থান দুটিতে পদার্থ১০৩ কোর্সের যথাক্রমে মিডটার্ম ও ফাইনাল পরীক্ষার নম্বর রাখতে পারি। অতপর একই পদ্ধতিতে মান দুটি নিয়ে যোগ করে তার ফল প্রকাশ করতে পারি। এই জাতীয় কাজ করার জন্য ডাটা টাইপ ও ভেরিয়েবল সম্পর্কে ধারণা থাকা দরকার।

ডাটা টাইপ ও ভেরিয়েবল:

যেহেতু রান করার আগে ব্যবহারকারী কি ইনপুট সিঙ্কল দিতে পারে তা আমাদের প্রোগ্রাম আগে থেকে জানে না এবং রান করার পর আমাদের প্রোগ্রাম ইউজারের নিকট হতে তা চেয়ে নিবে সেক্ষেত্রে একটি ভেরিয়েবল (বাংলায় বলে চলক) লাগবে যা প্রোগ্রাম রান করার পর কিবোর্ড হতে পাওয়া ইনপুট সিঙ্কলকে ধরে রাখবে। ধরে নেই যে চলকটি হল symbol। এখন কিবোর্ড হতে প্রদেয় সিঙ্কলটি হতে পারে একটি নম্বর (যেমন -১০ ১২, ইত্যাদি) বা একটি কেবোর্ড (যেমন a, d,x,z, ইত্যাদি)। ফলে সি প্রোগ্রাম যে সকল ডাটা প্রসেস করে তার টাইপ সম্পর্কে প্রোগ্রামকে বলে বুঝিয়ে দিতে হয়। এর চারটি প্রধান ডাটা টাইপ হল:

1. integer যার ভেলু রেঞ্জ হল -32,768 to 32,767 যে কোন পূর্ণ সংখ্যা এবং এর সি ডিক্লারেশন পদ্ধতি হল int
 2. floating point যার ভেলু রেঞ্জ হল 1.2E-38 to 3.4E+38 যেকোন ভগ্নাংশ সংখ্যা এবং এর সি ডিক্লারেশন পদ্ধতি হল float
 3. double যার ভেলু রেঞ্জ হল 2.3E-308 to 1.7E+308 যে কোন ভগ্নাংশ সংখ্যা এবং এর সি ডিক্লারেশন পদ্ধতি হল double
 4. character যা কিবোর্ডের যে কোন একটি কেবোর্ডকে বুঝায় এবং এর সি ডিক্লারেশন পদ্ধতি হল char
- কাজেই বুঝা গেল যে আমাদের প্রোগ্রামে ইন্টিজার টাইপের কোন মান নিয়ে কাজ করতে চাইলে প্রথমেই ভেরিয়েবল ডিক্লারেশন করে নিতে হবে। ডিক্লারেশনের পদ্ধতি হল:

int Marks;

যেখানে একটি মাত্র ভেরিয়েবল Marks যার ভিতর -32,768 to 32,767 এর মধ্যে যে কোন মান রাখা যাবে।

আবার একাধিক ভেরিয়েবল হলে ভেরিয়েবলগুলিকে কমা দিয়ে পৃথক করতে হয়। যেমন:

int MidTerm, Final, TotalMarks;

এখানে MidTerm, Final এবং TotalMarks প্রত্যেকেই এক একটি ইন্টিজার টাইপের ভেরিয়েবল।

ভেরিয়েবলের নাম যে কোন কিছু হতে পারে তবে তা একটি কেবোর্ড দিয়ে শুরু হতে হবে।

একই পদ্ধতিতে floating point এর জন্য হতে পারে

float Marks1, Marks2, TotalMarks;

অর্থাৎ প্রথম ডাটা টাইপের জন্য কিওয়ার্ড লিখতে হবে। তারপর ভেরিয়েবলগুলিকে কমা দিয়ে পৃথক করতে হবে।

শেষে সেমিকোলন বসাতে হবে। যেমন:

double Marks1, Marks2, TotalMarks;

অথবা

char grade;

মনে রাখতে হবে যে প্রোগ্রামের একই অংশের ভিতর একই নামে একাধিক ভেরিয়েবল ডিক্লারেশন করা যাবে না।

কিবোর্ড হতে ভেরিয়েবলে মান গ্রহণ করার কমান্ড হল:

1. ইন্টিজার টাইপের ডাটা পড়ার জন্য লিখতে হবে: `scanf("%d",&variale's name);`

যেমন:

```
int Marks;  
scanf("%d",&Marks);
```

আবার Marks এর মান আউটপুটে পাঠাতে লিখতে হবে

```
printf("%d",Marks);
```

2. ফ্লট টাইপের ডাটা পড়ার জন্য লিখতে হবে: `scanf("%f",&variale's name);`

যেমন:

```
float GradePoint;  
scanf("%f",&GradePoint);
```

GradePoint এর মান আউটপুটে পাঠাতে লিখতে হবে

```
printf("%f",GradePoint);
```

3. ডবল টাইপের ডাটা পড়ার জন্য লিখতে হবে: `scanf("%lf",&variale's name);`

যেমন:

```
double profitRate;  
scanf("%lf",&profitRate);
```

profitRate এর মান আউটপুটে পাঠাতে লিখতে হবে

```
printf("%lf",profitRate);
```

4. কেরেক্টার টাইপের ডাটা পড়ার জন্য লিখতে হবে: `scanf("%c",&variale's name);`

char letter;

```
scanf("%c",&letter);
```

letter এর মান আউটপুটে পাঠাতে লিখতে হবে

```
printf("%c",letter);
```

চলকের মান দিয়ে ত্রিভুজ ড্র করা:

আজকের টিউরিয়ালের উপরে যে প্রোগ্রামটি আছে আমরা যদি সেখানে প্রতিটি * কে অন্য কোন কেরেক্টার দিয়ে পরিবর্তন করতে পারি তাহলে সেই প্রোগ্রাম * এর পরিবর্তে আমাদের দেওয়া কেরেক্টার দিয়ে ত্রিভুজ আঁকবে। সেক্ষেত্রে আমাদের দেওয়া কেরেক্টারকে শুরুতে `scanf` ফাংশন দিয়ে কোন একটি ভেরিয়েবলে নিয়ে নিতে হবে। ধরি সেই ভেরিয়েবলটির নাম হল `symbol` যার টাইপ হল `char`। এই ক্ষেত্রে আমরা কিবোর্ড হতে যে কেরেক্টার টাইপ করব তা `symbol` নামক ভেরিয়েবলে সেভ হবে। এবার আমরা উপরের প্রোগ্রামের কমান্ডের ভিতরে প্রতিটি * এর জায়গায় লিখব `%c` ও শেষে লিখব সমান সংখ্যক `symbol`, তাহলে তা আমাদের `symbol`কে আউটপুটে পাঠাবে। যেমন `printf("***\n");` এর জায়গায় লিখব `printf("%c%c%c",symbol,symbol,symbol);` তখন যদি আমরা `a` চাপি তাহলে উক্ত কমান্ড আউটপুটে দেখাবে `aaa`। তাহলে প্রোগ্রামটির দুটি অংশ হবে। একটি অংশ হলো ইনপুট নেবার জন্য। অপর অংশ হলে কিবোর্ড হতে গ্রহণ করা সিঙ্গেল দিয়ে ত্রিভুজটি আঁকা।

এই পদ্ধতিতে প্রোগ্রামটি হল নিম্নরূপ:

```
#include<stdio.h>  
#include<stdlib.h>  
#include<conio.h>
```

```

void main()
{
char symbol;
printf("Enter a symbol\n");
scanf("%c",&symbol);

printf("%c\n", symbol);
printf("%c%c%c\n", symbol, symbol, symbol);
printf("%c%c%c%c%c\n", symbol, symbol, symbol, symbol, symbol);
printf("%c%c%c%c%c%c%c%c\n", symbol, symbol, symbol, symbol, symbol, symbol, symbol, symbol);
printf("%c%c%c%c%c%c%c%c%c%c\n", symbol, symbol, symbol, symbol, symbol, symbol, symbol, symbol, symbol, symbol);
getch();
}

```

উদাহরণ:

এখন আমরা ধরে নেই যে রহিম একজন বিশ্ববিদ্যালয় পড়ুয়া ছাত্র। সে এ মাসে বাড়ি থেকে নিয়ে আসল ৫০০০ টাকা। সে টিউশনি করে আয় করল ২০০০ টাকা। তাহলে এ মাসে সে মোট কত টাকা খরচ করার সুযোগ পেল? তা বের করতে হবে। নিচে তার জন্য একটি প্রোগ্রাম দেওয়া হলো।

আমাদের যা করতে হবে তাহল যে, তার বাড়ি থেকে নিয়ে আসা টাকা একটি ভেরিয়েবলে রাখব। ধরে নেই সে ভারিয়ারের নাম হলো fromFather। একইভাবে টিউশনির টাকার পরিমাণ কম্পিউটারের মেমরিতে ধরে রাখার জন্য ভেরিয়েবল হলো fromTution। এই দুই ভেরিয়েবলের মান যোগ করে তৃতীয় ভেরিয়েবল Total এ রাখব এবং Total এর মান আউটপুটে পাঠাব। তাহলেই কাজ শেষ। এখানে fromFather, fromTution এবং Total প্রত্যেকেই ইন্টিজার টাইপ ভেরিয়েবল। উপরে বর্ণিত রহিমের মোট টাকা বের করতে হলে fromFather এর মধ্যে ৫০০০ ও fromTution এর ভিতর ২০০০ রাখতে হবে, অর্থাৎ fromFather=5000; এবং fromTution=2000; বসাতে হবে। পরবর্তীতে লিখতে হবে Total=fromFather+fromTution;

```

#include<stdio.h>
#include<conio.h>
void main()
{
int fromFather, fromTution, Total;
fromFather=5000;
fromTution=2000;
Total=fromFather+fromTution;
printf("Rahim now has %d Taka", Total);
getch();
}

```

উপরের উদাহরণে দেখলাম যে রহিম তার বাবার নিকট হতে প্রতিমাসে ৫০০০ টাকা ও টিউশনি হতে ২০০০ টাকা করে পায়। এ দুটি মানই নির্ধারিত। কিন্তু যদি টাকার পরিমাণ নির্ধারিত না হয়, অর্থাৎ পরিবর্তনশীল, তাহলে উপরের প্রোগ্রাম কাজ করবে না। প্রতিবার প্রোগ্রাম রান করার আগে প্রোগ্রামের ভিতরে এ দুই ভেরিয়েবলের মান পরিবর্তন করতে হবে। যা জটিল, কেননা তার জন্য মূল প্রোগ্রামে পরিবর্তন করতে হচ্ছে। এখন বিকল্পটি বুঝতে ধরে নেই যে এই মানগুলি এক এক মাসে এক এক রকম হতে পারে। তাহলে রান করার পর যদি কিবোর্ড হতে মান নেওয়া যায়, তবে একই প্রোগ্রাম প্রতিমাসের হিসাব বের করতে কাজ করবে ফলে আমরা দুই ভেরিয়েবলের মান রান করার পর কিবোর্ড হতে নিব। অতপর দুই মান যোগ করে তা আউটপুটে প্রকাশ করব। তখন প্রোগ্রামটি হবে নিম্নরূপঃ

```
#include<stdio.h>
#include<conio.h>
void main()
{
int fromFather, fromTution, Total;
printf("Enter the amount that Rahim got from his father");
scanf("%d", &fromFather);
printf("Enter the amount that Rahim got from his tution");
scanf("%d", &fromTution);
Total=fromFather+fromTution;
printf("Rahim now has %d Taka", Total);
getch();
}
```