

সি প্রোগ্রামিং সহজ পাঠ

মুহম্মদ জাহিদুল হক

যন্ত্রযোগ প্রোগ্রামিং ক্লাবের ফেসবুক গ্রুপে প্রকাশিত সি প্রোগ্রামিং বিষয়ক ধারাবাহিক পোস্ট সমূহের পরিমার্জিত
সংকলন

পোস্ট -০১

কোড ব্লক টা কি তোমার কম্পিউটারে ইন্সটল করা আছে? থাকলে তো ভাল না থাকলে

<http://sourceforge.net/.../codeblocks-13.12mingw-set.../download> এই লিংক থেকে ডাউনলোড করে ইন্সটল করে ফেল।

ইন্সটল করা আছে? তাইলে কোডব্লক ওপেন করে Ctrl+Shift+N চাপ দেখবে একটা কোড লেখার জায়গা আসছে। এইবার নিচের কোডটা লিখে ফেল--

```
#include<stdio.h>
int main()
{
printf("Hello! this is your first C Program");
return 0;
}
```

এইবার ফাইলটা সেভ কর। যেকোন একটা নাম দিতে পার। ধরে নিলাম এইটা তোমার প্রথম প্রোগ্রাম নাম দিতে পার First. Save as type টা দেখে নিও তো C/C++ Files করা আছে নাকি। করা থাকলে তো ভাল না থাকলে C/C++ Files টাইপ করে ফেল।

এখন F9 চাপ। কী একটা কালো উইন্ডো এসেছে? এটাকে বলে কনসোল। এইখানে লেখা এসেছে Hello! this is your first C Program

পরের পোস্টে কোডটা বোঝানোর চেষ্টা করব। এখন যারা নতুন তারা উপরের কাজগুলো করে ফেল।

পোস্ট-০২

তোমাগো আগের পোস্টে কোডব্লকস ব্যবহার কইরা কি কইরা সি কোডিং করা যায় তা দেখাইছিলাম। আশা করি যারা আগ্রহী তারা চেষ্টা কইরা দেখছ। আমাদের আগের পোস্টের কোডটা ছিল-

```
#include<stdio.h>
int main()
{
printf("Hello! this is your first C Program");
return 0;
}
```

সি প্রোগ্রাম এক বা একাধিক ফাংশন নিয়ে তৈরী। এর মধ্যে main() ফাংশনটা অবশ্যই থাকতে হইব। তোমাগো ফাংশন চেনার একটা সহজ উপায় বইলা দেই, যেই শব্দের শেষে প্রথম বন্ধনী "()" দেখবা সেইটা ফাংশন। এখন এই বন্ধনীর ভিতর কিছু থাকতেও পারে আবার নাও থাকতে পারে।

আইচ্ছা এখন আমারে কও তো এই প্রোগ্রামে ফাংশন আছে কয়টা?

পোস্ট-০৩

তাইলে আগের পোস্ট থেকে তোমরা দুইটা ফাংশন চিনতে পারছ। একটা হইলো main() আর একটা হইল printf()। আগের পোস্টে বলছিলাম যে main() ফাংশনটা সি প্রোগ্রাম এ থাকতেই হইব। আর printf() ফাংশন দিয়া যখন তুমি কোন লেখা মনিটরে আউটপুট হিসাবে পাইতে চাও তখন ব্যবহার করবা।

printf() ফাংশনটা হইল একটা লাইব্রেরী ফাংশন। আর এই ফাংশনটা যদি তুমি তোমার প্রোগ্রামে ব্যবহার করতে চাও তাইলে main() ফাংশন এর উপরে #include<stdio.h> লাইনটা ব্যবহার করতে হইব। stdio.h হইল হেডার ফাইল যেখানে printf() ফাংশনটা ক্যামনে কাজ করব হের নীল নকশা দেওয়া আছে। আর এই হেডার ফাইলটা তোমার প্রোগ্রামে যোগ করার জন্য #include হইল প্রিপ্রেসেসর ডাইরেক্টিভ নামধারী একটা জিনিস।

হেনরী রাইডার হ্যাগার্ড এর শী বইটা পড়ছ? সেবা প্রকাশনী থেইকা এইটার একটা অনুবাদ বাইর হইছিল। চাঙ্গ পাইলে পইড়া ফালাইও।

পোস্ট-০৪

তোমরা তো আশা করি **printf()** ফাংশনটা কেমনে সি প্রোগ্রামে ব্যবহার করবা হেইডা বুঝতে পারছ। চল আরেকটা কোড দেখি-

```
#include <stdio.h>
int main()
{
printf("My Name is Zahid.");
printf("I am a student of HSC.");
return 0;
}
```

উপরের কোডটা দেখ আমি এইখানে **printf()** ফাংশনটা দুইবার কল করছি দুইটা বাক্য মনিটরে দেখানোর জন্য। তার মানে আমি কম্পিউটারকে দুইটা কাজের নির্দেশ দিছি। প্রত্যেকটা নির্দেশ এর পর আমি সেমিকোলন দিছি। সি প্রোগ্রামে **প্রত্যেকটা নির্দেশকে বলে statement**, আর স্টেটমেন্ট শেষ করতে হয় সেমিকোলন দিয়া। আমি উপরের কোডটা লেখছিলাম এই চিন্তা কইরা যে দুইটা বাক্যের যাতে কম্পিউটার ব্যাটা দুই লাইনে প্রকাশ করে কিন্তু সে আমার মনের কথাটা বুঝতে পারল না। চিন্তা কর কত বড় গবেষ্টা!!! এই ব্যাটারে আবার **"\n"** লেইখা **নতুন লাইন** বোঝানো লাগে। তাই বাধ্য হইয়া কোডটা পরিবর্তন করতে হইল-

```
#include <stdio.h>
int main()
{
printf("My Name is Zahid.\n");
printf("I am a student of HSC.");
return 0;
}
```

দুইটা কোডই তোমরা ট্রাই কইরা দেইখ। আর শোন খিলগাঁও তালতলা মার্কেটে যে সাত লেয়ারের চা পাওয়া যায় হেইডা কি তোমরা জান? চাম্প পাইলে ট্রাই মাইরা দেইখ। আমার মতে অতি অখাদ্য।

পোস্ট-০৫

মনে কর তুমি একটা প্রোগ্রাম বানাইতে চাও যেখানে তুমি দুইটা পূর্ণসংখ্যা কম্পিউটারকে দিবা আর সে সেইটা প্রসেস কইরা তোমারে যোগফল দেখাইব। দেখ ভাই পূর্ণসংখ্যা কারে বলে এইটা এখন বুঝাইতে পারমু না। তোমাগো গনিত বইয়ে থাকার কথা। এখন তাইলে আমাদের সমস্যাটায় আসি। যেহেতু আমরা দুইটা সংখ্যা দিমু তাইলে কম্পিউটারের মেমরিতে দুইটা ঘর খালি করতে হইব যেথারে পূর্ণসংখ্যা রাখা যাইব আর একটা ঘর খালি রাখতে হইব যেখানে যোগফলটা সংরক্ষন করতে হইব। এইটা আমরা কম্পিউটারকে সি প্রোগ্রামের মাধ্যমে কিকইরা নির্দেশ দিমু?

```
int num1, num2, sum;
```

উপরের লাইনটা লেখলে কম্পিউটার বুঝব **int** মানে **Integer** টাইপের ডেটা রাখার জন্য **num1**, **num2** আর **sum** নামের তিনটা ঘর খালি রাখতে হইব। এই **num1**, **num2**, **sum** এইগুলোকে সি প্রোগ্রামিং এ বলে **Identifier**.

```
num1=10;
num2=12;
```

উপরের দুইটা লাইন দিয়া কম্পিউটারেরে বলা হইল যে **num1** ঘরে **10** রাখ আর **num2** ঘরে **12** রাখ। এই খানে একটা কথা বইলা রাখি, আমরা সাধারনত জানি যে **a=b** মানে হইল **a** আর **b** এর মান সমান কিন্তু সি প্রোগ্রামিং এ এই কথাটার মানে হইল **a** ঘরে **b** এর মান বসব।

```
sum=num1+num2;
```

এই লাইন দিয়া বুঝাইলাম যে num1 আর num2 ঘরের ডেটাগুলো যোগ কইরা sum ঘরে রাখতে কইছি।

তোমরা তো জান যে আমরা **printf()** ফাংশনটা ব্যবহার করি আউটপুট দেখানোর লাইগ্যা। এখন দেখামু কেমনে তুমি **Identifier** এর মান মনিটরে দেখাইবা। এই খানে আমরা sum ঘরের মানটা দেখামু।

```
printf("The sum is %d",sum);
```

এখানে "%d" এর জায়গায় দেখাইব sum ঘরের মান। তুমি যদি **Integer** মান দেখাইতে চাও তাইলে "%d" ব্যবহার করবা। তাইলে চল দেখি যদি পুরা প্রোগ্রামটা লিখি তাইলে কেমন দেখাইব-

```
#include<stdio.h>
int main()
{
    int num1, num2, sum;
    num1=10;
    num2=12;
    sum=num1+num2;
    printf("The sum is %d",sum);
    return 0;
}
```

একটু ঝামেলা লাগতাত্বে? আস্তে আস্তে ঠিক হইয়া যাইব। আর ইফতারের সময় বেশি কইরা পানি খাইও, ভাজাপোড়া কম খাইও।

পোস্ট-০৬

আমাদের সর্বশেষ পোস্টের প্রোগ্রামটা ছিল -

```
#include<stdio.h>
int main()
{
    int num1, num2, sum;
    num1=10;
    num2=12;
    sum=num1+num2;
    printf("The sum is %d",sum);
    return 0;
}
```

তোমাদের আমি **Identifier** সম্পর্কে জানাইছি, এই পোস্টে আরেকটা নতুন নাম শিখাব। সেইটা হইল **Operator**. খেয়াল কর আমরা প্রোগ্রামের মধ্যে দুই বা তার বেশি **Identifier** এর মধ্যে সম্পর্ক বোঝানোর জন্য =, + এই দুইটা চিহ্ন ব্যবহার করছি। এদেরকে সি প্রোগ্রামে বলে **Operator**. এই দুইটা ছাড়াও আরো অপারেটর আছে যাদের সাথে আমাদের খুব শিগ্ৰই পরিচয় হবে।

আজকে আমরা আগের সমস্যাটার সাথে আরেকটা নতুন সমস্যা যোগ করব। এইবার আমরা আমাদের প্রোগ্রামের ইউজার এর কাছ থেকে ডেটা নিব। তার মানে হইল ইউজার তার ইচ্ছামত সংখ্যা দিব আর আমরা প্রোগ্রাম সেইটা যোগ করব।

ইউজারের কাছ থিক্যা ডেটা নেওয়ার লাইগ্যা আমরা একটা ফাংশন ব্যবহার করি সেইটা হইল **scanf()** ।

```
scanf("%d",&num1);
```

এই লাইনের মাধ্যমে আমরা কম্পিউটারকে নির্দেশ দিলাম যে তুমি একটা **Integer** মান ইউজারের কাছ থেকে ইনপুট নিয়া **num1** ঘরে রাখ। তাইলে আমরা যদি আগের প্রোগ্রামটার দুইটা লাইন বদলাইয়া নিচের মত লেখি -

```
#include<stdio.h>
int main()
{
    int num1, num2, sum;
    printf("Enter The First Number:");
    scanf("%d",&num1);
    printf("Enter The Second Number:");
    scanf("%d",&num2);
    sum=num1+num2;
    printf("The sum is %d",sum);
    return 0;
}
```

প্রোগ্রামটা রান কইরা দেখতো ফলাফল কি দেয়। আর একটা কথা **scanf()** ও একটা লাইব্রেরী ফাংশন আর এই ফাংশনের হেডার ফাইল **stdio.h**

পোস্ট-০৭

আমাদের প্রতিটা প্রোগ্রামে তোমরা হয়ত খেয়াল করছ যে আমি **main()** ফাংশনের আগে **int** লেখতাম আর প্রতিটা প্রোগ্রাম শেষ করতাম নিচের স্টেটমেন্টটা লেইখা-

```
return 0;
```

প্রতিটা ফাংশনের একটা রিটার্ন আছে। আমরা **main()** ফাংশনেরও একটা রিটার্ন থাকা জরুরী। আমরা এইখানে **int** ব্যবহার কইরা ঠিক কইরা দিলাম যে রিটার্ন এর ধরনটা অবশ্যই ইন্টিজার টাইপের হইতে হইবে। এখন এই রিটার্ন মান টা যদি **main()** ফাংশন পায় তাইলে সে মনে করবে প্রোগ্রামটা সঠিকভাবে সম্পন্ন হইছে এবং তার কাজ শেষ। আমরা সর্বশেষে **return 0;** স্টেটমেন্টটা ব্যবহার কইরা **main()** ফাংশনের জানাইয়া দেই, এই লও তোমার একটা ইন্টিজার মান **0**। আর আমরা এই স্টেটমেন্ট টা যদি নির্বাহ হয় তবে প্রোগ্রামটা কোন সমস্যা ছাড়াই শেষ হইছে এটা নিশ্চিত।

পোস্ট-০৮

আজকে কথা বলমু আইডেন্টিফায়ার নিয়া। যদি তুমি কোন আইডেন্টিফায়ার এর মান **Fixed** কইরা দেও তাইলে সেইটারে বলে **Constant** আর যদি আইডেন্টিফায়ার এর মান প্রোগ্রামে পরিবর্তন হয় তাহলে সেইটারে বলে **Variable**। আমরা আমাদের আগের যোগ করার প্রোগ্রামে যে আইডেন্টিফায়ার গুলো ব্যবহার করছি সেইগুলো **Variable** (কারণ এখানে ব্যবহারকারী তার ইচ্ছামত মান দিতেছিল)। সি প্রোগ্রামে **Constant** ব্যবহার করার জন্য আমি নিচের পদ্ধতিটা ব্যবহার করি।

```
#define PI 3.14
```

উপরের লাইনে **#define** প্রিপ্রেসেসর ব্যবহার কইরা **PI** নামের আইডেন্টিফায়ার এর একটা ফিক্সড মান দেওয়া হইছে। এই লাইনে আমরা সেমিকোলন ব্যবহার করব না এবং কোন ডেটা টাইপ উল্লেখ করার দরকার নাই। অহন চল **constant** ব্যবহার কইরা ব্লকের স্কেফোল বের করার জন্য একটা সি প্রোগ্রাম বানাই।

```
#include<stdio.h>
#define PI 3.14
int main()
{
```

```
float radius, area;
printf("Enter the radius of the Circle:");
scanf("%f",&radius);
area=PI*radius*radius;
printf("Area of the circle: %f",area);
return 0;
}
```

প্রোগ্রামটা রান কইরা দেইখ। প্রোগ্রামটার বিস্তারিত ব্যাখ্যা পরের পোস্টে দিব। রাত জাইগা ফেসবুক চালানোটা ঠিক না। এইটা বাদ দিতে চেষ্টা কর।

পোস্ট-০৯

আমরা আগের পোস্টে বৃত্তের ক্ষেত্রফল বের করার একটা প্রোগ্রাম দিছিলাম। প্রোগ্রামটা ছিল-

```
#include<stdio.h>
#define PI 3.14
int main()
{
float radius, area;
printf("Enter the radius of the Circle:");
scanf("%f",&radius);
area=PI*radius*radius;
printf("Area of the circle: %f",area);
return 0;
}
```

আজকে আস এই পোস্টটার একটা ময়না তদন্ত করি আর নতুন কিছু জানা যায় কিনা দেখি। কোন বৃত্তের ক্ষেত্রফল বের করতে হলে আমাদের দুইটা মান দরকান হয় - বৃত্তের Radius আর পাই এর মান। বৃত্তের Radius হইল একটা চলক(Variable) কারণ আমরা ইউজার তার ইচ্ছামত এই মানটা দিব আর পাই এর মানতো স্থির (আমরা সাধারনত ৩.১৪ কেই এর মান ধরে নেই)। তাইলে আমরা

```
#define PI 3.14
```

উপরের লাইন দ্বারা পাই এর মান স্থির কইরা রাখলাম। যেহেতু হের মান পরিবর্তন হইবো না।

```
float radius, area;
```

উপরের লাইন দিয়া আমরা কম্পিউটারে দুইটা জায়গা খালি রাখতে বললাম যেখানে আমরা ভগ্নাংশযুক্ত সংখ্যা রাখব। **float** কীওয়ার্ডটা ভগ্নাংশযুক্ত ডেটাটাইপ ঘোষনা করার জন্য ব্যবহার হয়।

```
area=PI*radius*radius;
```

তোমরা তো জান **বৃত্তের ক্ষেত্রফল = পাই X Radius X Radius**, সি প্রোগ্রামিং এ গুন (X) চিহ্নটারে (*) স্টার চিহ্নদ্বারা প্রকাশ করা হয়। তাইলে আমরা উপরের লাইন দ্বারা **area** নামের Identifier এর মধ্যে বৃত্তের ক্ষেত্রফল পামু।

পোস্ট-১০

scanf() কিন্বা **printf()** ফাংশনে মাধ্যমে যখন আমরা কোন Identifier মানটা গ্রহন কিন্বা প্রদর্শন করতে চাই তখন ঐ Identifier কোন ধরনের ডেটা ধারণ করে তার উপর ভিত্তি করে **ফরম্যাট স্প্যাসিফায়ার** দিতে হয়। যেমন যদি

Identifier এর মান **int** টাইপের হয় তাহলে আমরা ব্যবহার করতেন **%d** আর যদি **float** টাইপের হয় তাহলে ব্যবহার করতেন **%f**, তোমাদের সুবিধার্থে আজকে কোন টাইপের জন্য কোন ফরম্যাট স্প্যাসিফায়ার দিবা তার একটা চার্ট দিয়ে দিলাম।

ch এর জন্য %c
String এর জন্য %s
float এর জন্য %f
int এর জন্য %d
double এর জন্য %lf
int এর মান হেক্সাডেসিমাল আকারে প্রকাশ করতে %x
int এর মান অক্টাল আকারে প্রকাশ করতে %o

আস আমরা একটা প্রোগ্রামের মাধ্যমে পুরা কাজটা আবার রিভিশন দেই-

```
#include <stdio.h>
int main()
{
    char ch = 'A';
    char str[20] = "Zahidul Huq";
    float flt = 10.234;
    int no = 150;
    double dbl = 20.123456;
    printf("Character is %c \n", ch);
    printf("String is %s \n", str);
    printf("Float value is %f \n", flt);
    printf("Integer value is %d\n", no);
    printf("Double value is %lf \n", dbl);
    printf("Octal value is %o \n", no);
    printf("Hexadecimal value is %x \n", no);
    return 0;
}
```

উপরের প্রোগ্রামের কিছু লাইন তোমার বোধগম্য নাও হতে পারে। চিন্তা কইরো না, আমরা বিষয়গুলো সামনে জানতে পারব।

পোস্ট-১১

তোমরা আগের প্রোগ্রামগুলোতে খেয়াল করবা যে সি প্রোগ্রামে স্টেটমেন্টগুলো একটার পর এর একটা সিরিয়ালি এক্সিকিউট হইতেন। আমাদের মাঝে মধ্যে দরকার পরবে যে শর্তসাপেক্ষে কোন স্টেটমেন্ট কাজ করবে কিনা কোন স্টেটমেন্ট একাধিক বার কাজ করবে কিনা এটা নির্ধারণ করে দেওয়া। সি প্রোগ্রামিং এ এই সুবিধা পাওয়ার জন্য **কন্ট্রোল স্টেটমেন্ট** নামক একটা জিনিষ আছে। আজকে আমরা একটা কন্ট্রোল স্টেটমেন্ট এর ব্যবহার শিখব।

ধর আমরা কোন ইউজার এর কাছ থেকে একটা পূর্ণসংখ্যা নিলাম। এখন প্রোগ্রাম আউটপুট দিবে যে সংখ্যাটা জোড় নাকি বিজোড়।

তাহলে আমাদের প্রোগ্রামের দুইটা স্টেটমেন্ট থাকবে আউটপুটের জন্য একটা যদি জোড় সংখ্যা হয় তবে সে দেখাবে **This is an even number** আর যদি বিজোড় সংখ্যা হয় তবে **This is a odd number**. চল প্রথমে প্রোগ্রামটা দেখি-

```
#include<stdio.h>
int main()
{
    int num;
    printf("Enter The Number:");
    scanf("%d",&num);
    if(num%2==0)
```

```
printf("This is an even number");
if(num%2!=0)
printf("This is a odd number");
return 0;
}
```

এইখানে আমরা *if(num%2==0)* এই লাইনটার মাধ্যমে বুঝাইলাম যদি *num%2* মানে *num* কে 2 দ্বারা ভাগ করলে ভাগশেষ, *==* মানে হইল ভাগশেষ যদি 0 এর সমান হয় তবে (*খয়াল কর এই স্টেটমেন্টের মধ্যে আমরা ";" সেমিকোলন ব্যবহার করি নাই কারণ কন্ট্রোল স্টেটমেন্টএ কোন সেমিকোলন হবে না।*) *printf("This is an even number");* লাইনটা কাজ করবে নতুবা কাজ করবে না।

অনুরূপ ভাবে *if(num%2!=0)* এই লাইনটা বুঝাইতেছে যদি ভাগশেষ 0 না হয় (কারণ *"!="* operator এর মানে হইল অসমান), তবে *printf("This is a odd number");* লাইনটা কাজ করবে।

তাইলে আজকে আমরা if নামের কন্ট্রোল স্টেটমেন্টটার ব্যবহার আশা করি বুঝতে পারছি।

পোস্ট-১২

সর্বশেষ পোস্টে আমি তোমাদের সাথে if কন্ট্রোল স্টেটমেন্ট নিয়ে আলোচনা করেছিলাম। আজকে আমি আলোচনা করব আরেকটা নতুন কন্ট্রোল স্টেটমেন্ট নিয়ে।

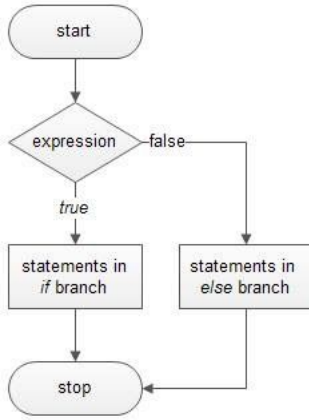
গত পোস্টে আমরা যে সমস্যা নিয়ে আলোচনা করেছিলাম সেটা আবার নিয়ে আসি - আমরা ইউজারের কাছ থেকে একটা পূর্ণ সংখ্যা নিব এরপর সেটা যদি জোড় সংখ্যা হয় তবে কম্পিউটার মনিটরে প্রিন্ট হবে - This is an even number আর যদি বিজোড় সংখ্যা হয় তবে মনিটরে প্রিন্ট হবে This is a odd number.

আমরা এই সমস্যাটা গত পোস্টে সমাধান করেছিলাম **if স্টেটমেন্ট** দিয়ে আর আজ করব **if....else স্টেটমেন্ট** দিয়ে। চল প্রথমে প্রোগ্রামটা দেখি-

```
#include<stdio.h>
int main()
{
int a;
printf("Enter the number:");
scanf("%d",&a);
if(a%2==0)
printf("This is an even number");
else
printf("This is a odd number");
return 0;
}
```

if(a%2==0) এই স্টেটমেন্টটা যদি সত্য হয় তবে if সংশ্লিষ্ট স্টেটমেন্ট ব্লকগুলো কাজ করবে নতুবা else সংশ্লিষ্ট স্টেটমেন্ট ব্লকগুলো কাজ করবে।

ছবিতে **if else কন্ট্রোল স্টেটমেন্ট** এর ক্লোচার্টটা সংযুক্ত করলাম—
তোমাদের জন্য ভালবাসা রইলো...



পোস্ট-১৩

মনে কর আমরা ইউজারের কাছ থেকে একটা পূর্ণসংখ্যা নিব আর আমাদের প্রোগ্রাম যাচাই করবে এইটা কি ধনাত্মক, ঋনাত্মক নাকি শূন্য দেয়া হয়েছে।

যদি ইউজারের ইনপুট হয় a , তাহলে আমাদের সম্ভাব্য আউটপুট হবে তিনটি

$if(a < 0)$ তবে প্রিন্ট হবে *This is a negative number*

$if(a > 0)$ তবে প্রিন্ট হবে *This is a positive number*

যদি উপরের কোন শর্তই পূর্ণ না করে তবে প্রিন্ট হবে *Number is zero*,

এখন তোমাদের কাজ হলো শুধু মাত্র `if` কন্ট্রোল স্টেটমেন্ট ব্যবহার করে একবার এই প্রোগ্রামটা তৈরী করবা এবং আরেকবার `if` এবং `if.....else` ব্যবহার করে প্রোগ্রাম তৈরী করবা।

আচ্ছা এতদিন ধইরা প্যাচাল পারতামি.... আজকে একটা পরীক্ষা নিতে পারমু না।

পোস্ট-১৪

গত পোস্টে একটা ছোট সমস্যা দিছিলাম। সমস্যাটা ছিল আমরা ইউজারের কাছ থেকে একটা পূর্ণসংখ্যা নিব আর আমাদের প্রোগ্রাম যাচাই করবে এইটা কি ধনাত্মক, ঋনাত্মক নাকি শূন্য দেয়া হয়েছে।

যদি ইউজারের ইনপুট হয় a , তাহলে আমাদের সম্ভাব্য আউটপুট হবে তিনটি

$if(a < 0)$ তবে প্রিন্ট হবে *This is a negative number*

$if(a > 0)$ তবে প্রিন্ট হবে *This is a positive number*

যদি উপরের কোন শর্তই পূর্ণ না করে তবে প্রিন্ট হবে *Number is zero*,

এখন আমাদের কাজ হলো শুধু মাত্র `if` কন্ট্রোল স্টেটমেন্ট ব্যবহার করে একবার এই প্রোগ্রামটা তৈরী করা এবং আরেকবার `if` এবং `if.....else` ব্যবহার করে প্রোগ্রাম তৈরী করা।

চল `if` দিয়া তৈরী করি -

```

#include<stdio.h>
int main()
{
    int a;
    printf("Enter The Number:");
    scanf("%d",&a);
    if(a<0) printf("This is a negative Number");
    if(a>0) printf("This is a Positive Number");
    if(a==0) printf("Number is Zero");
}
    
```

```
return 0;
}
```

এখন এই সমস্যাটাই আমরা সমাধান করতে if আর if.....else কন্ট্রোল স্টেটমেন্ট দুইটাই ব্যবহার করব।

```
#include<stdio.h>
int main()
{
    int a;
    printf("Enter The Number:");
    scanf("%d",&a);
    if(a==0) printf("Number is Zero");
    if(a<0)
    printf("This is a Negative Number");
    else
    printf("This is a Positive Number");
    return 0;
}
```

আমি চেষ্টা করলাম সমস্যাটা দুই ভাবে সমাধান করার কিন্তু সমস্যা একটা থেকে গেল। তোমরা কি একটু দেখবা প্রোগ্রাম গুলার মধ্যে কোথায় সমস্যা কিনা কোন প্রোগ্রামটার মধ্যে সমস্যাটা আছে?

দুই ধরনের প্রেমিক/প্রেমিকা আছে ... একদল অন্যের মনে ইর্ষা ধরানোর জন্য প্রেম করে, আরেকদল ভালবাসে বলেই ভালবাসে। প্রোগ্রামিং এর প্রথম টাইপের প্রেমিকদের আমি প্রোগ্রামারই মনে করি না...

পোস্ট-১৫

আমাদের গত পোস্টে আমি একই সমস্যার জন্য দুটো প্রোগ্রাম দিয়েছিলাম যাদের গঠন প্রকৃয়া ছিল ভিন্ন। তোমরা যারা কোডগুলো নিয়া কাজ করছ খেয়াল করবা প্রথম প্রোগ্রামটা ফলাফল দিবে একদম ঠিক ঠিক কিন্তু দ্বিতীয় প্রোগ্রামে যদি শূন্য ইনপুট দাও তাহলে সে এইটাকে ধনাত্মক সংখ্যা হিসাবে এবং Zero হিসাবে বিবেচনা করবে। তার মানে হইল প্রোগ্রামটাতে Syntax জাতীয় কোন সমস্যা না থাকলেও উত্তরটা সঠিক নয়। কারন শূন্য ধনাত্মক কিনা ঋণাত্মক হতে পারে না।

তাইলে আমরা কি এই সমস্যাটা কাটানোর একটা ব্যবস্থা নিতে পারি না? চল আমরা প্রোগ্রামটা নতুন করে আবার তৈরী করি।

```
#include<stdio.h>
int main()
{
    int a;
    printf("Enter The Number:");
    scanf("%d",&a);
    if(a!=0)
    {
        if(a<0) printf("This is Negative Number.");
        if(a>0) printf("This is a Positive Number");
    }
    else
    printf("This is Zero");
    return 0;
}
```

আমাদের হাতে আরো কিছু কন্ট্রোল সেন্টমেন্ট আছে, তাদের ব্যবহার সামনে শিখব ইনশাল্লাহ। আপাতত এই দুটোকে আরো কিভাবে ব্যবহার করা যায় চিন্তা কর।

পোস্ট ১৬

আজকে আমরা একটা নতুন কন্ট্রোল স্টেটমেন্ট এর ব্যবহার শিখব এবং সেটা হলো **elseif**, এই কন্ট্রোল স্টেটমেন্টটা ব্যবহার করতে হয় **if....else** স্টেটমেন্ট এর ভিতরে। আস আমরা আমাদের গত পোস্টের সমস্যাটাই সমাধান করি **elseif** কন্ট্রোল স্টেটমেন্ট ব্যবহার করে, তাহলে হাতেকলমে দেখা হয়ে যাবে **elseif** এর ব্যবহার।

আমাদের গত পোস্টের সমস্যাটা ছিল আমরা ইউজারের কাছ থেকে একটা পূর্ণসংখ্যা নিব আর আমাদের প্রোগ্রাম যাচাই করবে এইটা কি ধনাত্মক, ঋনাত্মক নাকি শূন্য দেয়া হয়েছে।

যদি ইউজারের ইনপুট হয় a , তাহলে আমাদের সম্ভাব্য আউটপুট হবে তিনটি

$if(a < 0)$ তবে প্রিন্ট হবে *This is a negative number*

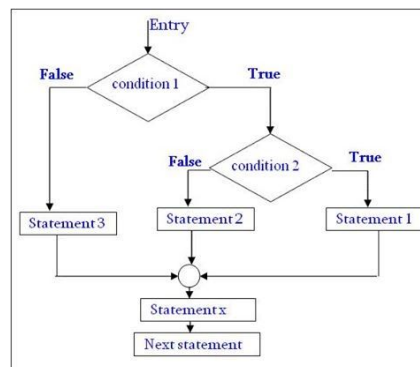
$if(a > 0)$ তবে প্রিন্ট হবে *This is a positive number*

যদি উপরের কোন শর্তই পূর্ণ না করে তবে প্রিন্ট হবে *Number is zero*

চল প্রোগ্রামটা দেখি-

```
#include<stdio.h>
int main()
{
    int a;
    printf("Enter The Number:");
    scanf("%d",&a);
    if(a<0)
        printf("This is a Negative Number");
    elseif(a>0)
        printf("This is a Positive Number");
    else
        printf("This is Zero");
    return 0;
}
```

এই কন্ট্রোল স্টেটমেন্টটার মাধ্যমে আমরা **if.....else** কন্ট্রোল স্টেটমেন্টের ভিতরে আরেকটা কন্ডিশন যোগ করতে পারি। **elseif** কন্ট্রোল স্টেটমেন্ট এর একটা প্রবাহ চিত্র পোস্টে সংযুক্ত করে দিলাম।



পোস্ট ১৭

কন্ডিশনাল কন্ট্রোল স্টেটমেন্টের মাধ্যমে আমরা কোন স্টেটমেন্ট কিম্বা স্টেটমেন্ট ব্লকের **Execution** নিয়ন্ত্রন করতে পারি বিভিন্ন শর্ত আরোপের মাধ্যমে। আমরা আজকে একটা নতুন সমস্যা সমাধান করব, আমরা মান হিসাবে ইউজারের কাছ থেকে তিনটি বাহুর মান নিব এবং আমাদের প্রোগ্রাম ফলাফল দিবে যে বাহু তিনটির সমন্বয়ে ত্রিভুজ গঠন সম্ভব কিনা।

আমার ধারণা আমরা এতদিনে এই সমস্যাটা সমাধান করার টেকনিক্যাল যোগ্যতা অর্জন করেছি, যদি আমাদের ধারণা থাকে কোন শর্তগুলো পূরণ করলে তিনটি বাহু ত্রিভুজ গঠন করে।

তিনটা বাহু ত্রিভুজ গঠন করবে যদি ,

১. কোন বাহুর মান শূন্য হতে পারবে না।

২. দুটো বাহুর যোগফল অবশ্যই তৃতীয় বাহু হতে ছোট হওয়া যাবে না।

চল প্রোগ্রামটা লিখে ফেলি-

```
#include<stdio.h>
int main()
{
    int a,b,c;

    printf("Enter The First Side:");
    scanf("%d",&a);

    printf("Enter The Second Side:");
    scanf("%d",&b);

    printf("Enter The Third Side:");
    scanf("%d",&c);

    if(a||b||c!=0 && a+b>c && b+c>a && a+c>b)
        printf("Sides will make triangle");
    else
        printf("Sides will not make triangle");

    return 0;
}
```

আমরা "||" অপারেটরটা ব্যবহার করি "অথবা" বুঝাতে আর "&&" অপারেটর ব্যবহার করি "এবং" বোঝানোর জন্য। আশা করি এই প্রোগ্রামের if স্টেটমেন্টটা তোমরা বুঝতে পারবা।

পোস্ট ১৮

এখন আমরা একটা প্রোগ্রাম বানাব যেটা ইউজারের কাছ থেকে ডেটা নিবে একটা চার ডিজিটের ইংরেজী সাল এবং আউটপুট দেখাবে যে সালটি লীপ ইয়ার কিনা।

যেহেতু ইউজার এর কাছ থেকে সালটা নিব তাই একটা Variable তো ঠিক করতেই হয়।

```
int year;
```

কোন সালকে লীপ ইয়ার হবার জন্য কিছু শর্ত পালন করতে হয়-

১. যদি $year \% 400 == 0$ হয় অথবা

২. যদি $year \% 100 != 0$ হয় এবং $year \% 4 == 0$ হয়।

তাহলে যদি আমরা এই দুটো কন্ডিশন প্রোগ্রামে দিয়ে দেই তবেই আমাদের প্রোগ্রাম সালটা লীপ ইয়ার কিনা তা পরীক্ষা করতে পারবে। আমি `elseif` কন্ট্রোল স্টেটমেন্টটা ব্যবহার করলাম।

```
#include <stdio.h>
int main()
{
    int year;
    printf("Enter a year to check if it is a leap year\n");
    scanf("%d", &year);
    if ( year%400 == 0)
        printf("%d is a leap year.\n", year);
    elseif ( year%100 != 0 && year%4 == 0 )
        printf("%d is a leap year.\n", year);
    else
        printf("%d is not a leap year.\n", year);
    return 0;
}
```

তোমরা কি ভিন্নভাবে একটু চেষ্টা করে দেখবা নাকি?

পোস্ট ১৯

আজকে আমরা কন্ডিশনাল অপারেটর নামে একটা অপারেটরের ব্যবহার দেখব। এই অপারেটরের ব্যবহার গঠনটা হলো-

Exp1?Exp2:Exp3;

এখানে প্রথম Exp1 দিয়ে একটা শর্তযুক্ত স্টেটমেন্ট বোঝানো হয়েছে যার সত্যতা থাকলে Exp2 কাজ করবে নতুবা Exp3 কাজ করবে।

আমরা আমাদের লীপইয়ার বিষয়ক সমস্যাটা এই কন্ডিশনাল অপারেটর দিয়ে সমাধানের চেষ্টা করব নাকি? চল করেই ফেলি -

```
#include<stdio.h>
int main()
{
    int year;
    printf("Enter The Year");
    scanf("%d",&year);
    year%400==0||(year%100!=0&&year%4==0)?printf("This is a Leap Year"):printf("This is not a Leap Year");
    return 0;
}
```

সহজ ভাষায় সি প্রোগ্রামিং পোস্ট ২০

ধর আমরা একটা প্রোগ্রাম তৈরী করব যেখানে "I Love Bangladesh" বাক্যটি আউটপুট হিসাবে প্রদর্শন করবে ইউজার যতবার প্রদর্শন করতে চায়।

তাহলে এখানে একটা নির্দিষ্ট কাজকে একাধিক বার নির্বাহ করার প্রয়োজনীয়তা তৈরী হয়েছে। কোন একটি স্টেটমেন্টকে শর্তসাপেক্ষে একাধিক বার সম্পাদন করার জন্য এক ধরনের কন্ট্রোল স্টেটমেন্ট সি প্রোগ্রামে ব্যবহৃত হয়। এদেরকে বলে লুপ কন্ট্রোল স্টেটমেন্ট।

বেশ কয়েক ধরনের লুপ কন্ট্রোল স্টেটমেন্ট আছে এই পোস্টে আমরা দেখব "for loop" এর ব্যবহার।

চল প্রথমে প্রোগ্রামটার কোডটা লিখে ফেলি, তারপর না হয় ব্যাখ্যা করা যাবে-

```
#include<stdio.h>
int main()
{
    int a;
    printf("Enter how many times you want to show: ");
    scanf("%d",&a);
    for(int counter=0;counter<a;counter=counter+1)
    printf("I Love Bangladesh\n");
    return 0;
}
```

আমরা যখন লুপ ব্যবহার করব তখন তিনটা বিষয় মাথায় রাখবা এটাকে 3C বলতে পার -

১. **Counter Declaration and Initialization** (যতবারই তুমি চালাও না কেন শুরু তো করতে হবে, এই গননা শুরু করার জন্য এই অংশটা)
২. **Condition**, কোন শর্ত যতক্ষণ পূর্ণ হবে হবে ততক্ষণ একটা লুপ চলবে
৩. **Counter Increment / Decrement**, তুমি যদি Counting 1 থেকে শুরু কর তবে Counter এর মান বাড়াবা আর যদি ১০০ থেকে পিছনের দিকে যেতে চাও হবে Counter এর মান কমাবা।

আমাদের কোডে for দিয়ে আমি লুপ স্টেটমেন্টটা লেখা শুরু করলাম, এখানে 3C হবে "()" প্রথম বন্ধনীর ভেতর। int counter=0 এর মাধ্যমে আমরা Counter Declare এবং এর প্রারম্ভিক মান শূন্য ঠিক করলাম। counter<a এর মাধ্যমে Condition দিলাম যতক্ষণ Counter এর মান a এর কম থাকবে ততক্ষণ লুপটা চলতে থাকবে এবং counter =counter+1 এর মানে হল আমাদের স্টেটমেন্টটা প্রতিবার নির্বাহ হবার পর counter এর মান একবার বাড়বে। আর অবশ্যই এই স্টেটমেন্ট এর পর কোন সেমিকোলন হবেনা।

আর এই স্টেটমেন্টটা আমরা printf("I Love Bangladesh\n"); স্টেটমেন্ট কে নিয়ন্ত্রণ করার জন্য ব্যবহার করেছি।

পোস্ট ২১

আমাদের গত পোস্টে আমরা লুপ নামক শব্দটা সম্পর্কে একটা ধারণা নিয়েছিলাম। আজকে আমরা লুপ কন্ট্রোল স্টেটমেন্ট গুলোর মধ্যে আরেকটা অন্যতম একটা প্রকৃয়া while loop এর ব্যবহারটা শিখব। তোমাদের আগেই বলেছিলাম 3C সম্পর্কে এবং কন্ট্রোল স্টেটমেন্ট এ তাদের কার্য প্রকৃয়া সম্পর্কে। আমাদের গত পোস্টের সমস্যাটা আজকে আমরা while loop দিয়ে সমাধান করতে চেষ্টা করব।

আমরা একটা প্রোগ্রাম তৈরী করব যেখানে "I Love Bangladesh" বাক্যটি আউটপুট হিসাবে প্রদর্শন করবে ইউজার যতবার প্রদর্শন করতে চায়।

চল প্রোগ্রামটা লিখে ফেলি--

```
#include<stdio.h>
int main()
{
    int limit;
    printf("Enter How Many times You Want to display:");
    scanf("%d",&limit);
    int a=0; //এই লাইনের মাধ্যমে আমরা Counter Declaration এবং Initialization করলাম
    while(a<limit) //এই লাইনে থাকবে Condition, while() এর মধ্যে
    {
        printf("I Love Bangladesh"); // এই স্টেটমেন্টটা Execute হবে
        a=a+1; //Counter Increment
    }
}
```

```
return 0;
}
```

"/" চিহ্নটার পর তুমি যা লিখবা সেটাকে বলে কমেন্ট, এটা Execute হবে না। কেন লাইনটা লিখলাম এটা পরে মনে রাখার জন্য বা নিজস্ব কোন নোটের জন্য এটার ব্যবহার হয়। তাই বলে নোট আমার মত বাংলায় লেইখ্যা ফলাইও না।

পোর্ট ২২

আমরা জানি যে সি প্রোগ্রামিং হলো একটা স্ট্রাকচার্ড ল্যাঙ্গুয়েজ। এখানে উপর থেকে নিচের দিকে স্টেটমেন্টগুলো নির্বাহ হয়। তবে আমরা আমাদের প্রয়োজনে চাইলে যে কোন স্টেটমেন্টে প্রোগ্রামকে নিতে পারি। এই ধরনের একটা পদ্ধতি হলো goto স্টেটমেন্ট। এই কাজটা আমরা শর্ত সাপেক্ষে করতে পারি আবার শর্ত ছাড়াও করতে পারি। তবে একটা ব্যাপার খেয়াল রাখতে হবে এটা যাতে কোন অসীম লুপ তৈরী করতে না পারে।

চল আমরা একটা প্রোগ্রামের মাধ্যমে goto স্টেটমেন্টটা বোঝার চেষ্টা করি। আমাদের আজকের সমস্যা হলো আমরা ব্যবহারকারীর কাছ থেকে দুটো সংখ্যা নিব আর আমাদের প্রোগ্রাম তাদের গ.সা.গু নির্ণয় করবে।

আসা করি গ.সা.গু ব্যাপারটা সম্পর্কে তোমাদের ধারণা আছে। গ.সা.গু কী এটা জানা না থাকলে প্রোগ্রামটা বোঝা একটু সমস্যা হলে হতে পারে।

```
#include<stdio.h>
int main()
{
    int a,b,c;
    printf("Enter First Number:");
    scanf("%d",&a);
    printf("Enter Second Number:");
    scanf("%d",&b);
    if(a<b)
        c=a;
    else
        c=b;
    again: //এইটা হলো goto স্টেটমেন্ট এর লেভেল
    if(a%c==0&&b%c==0)
        printf("The GCD is: %d",c);
    else
    {
        c=c-1;
        goto again; // সোজা again লেখা লেভেল এর পর থেকে আবার নির্বাহ শুরু করবে
    }
    return 0;
}
```

এইখানে আমরা শর্ত সাপেক্ষে goto স্টেটমেন্টটা ব্যবহার করলাম এবং again নামক লেভেল ব্যবহার করেছি। তুমি লেভেল নাম তোমার পছন্দ মত ব্যবহার করতে পার।