

এসো 'সি' শিখি

সি প্রোগ্রামিং ল্যাংগুয়েজ কম্পিউটার ভাষাতে যথেষ্ট সফল জগাতে সক্ষম হয়েছে। একটি সাধারণ এপ্লিকেশন প্রোগ্রাম থেকে শুরু করে এডভান্সড এডিটর অপারেটিং সিস্টেম এবং যথেষ্ট জটিল এপ্লিকেশন প্রোগ্রাম সি প্রোগ্রামিং সাহায্যে তৈরি করা যায়। বিশেষ করে ট্রাঙ্কাচার প্রোগ্রামিং এর জন্য এটি প্রোগ্রামারদের নিকট বেশ প্রয়োজনীয় হয়েছে।

সি প্রোগ্রামিং ল্যাংগুয়েজ ব্যাপক। কাজেই যত্ন পরিচয়ে এর পূর্ণাঙ্গ উপস্থাপন সম্ভব নয়। তবে যাদের বিভিন্ন প্যাকেজ এবং অপারেটিং সিস্টেমের টুকটাকি কমান্ড ব্যবহারের হাতেখড়ি ইতিমধ্যেই হয়েছে তারা যাতে অল্প সময়ের মধ্যে নিজেই সিস্টেম প্রোগ্রাম শিখতে পারেন সেজন্য মনুলা প্রোগ্রাম ব্যবহার করে সেখানি সজাওনা হয়েছে। মেটাডাটা ট্রান্স ভিক দিয়ে সি চালানো সম্ভব। তবে সর্বকিছুই জানাতে কষ্ট হবে। কারণ এটির জন্য মেমোরী খরচ দরকার। তাই হার্ডডিস্ক সংলগ্ন কম্পিউটার ব্যবহার করা উচিত। দুঃখের সঙ্গে আওতা টাচেসি সি এর উপর ভিত্তি করে পাঠশালা বিভাগের এ লেখা। প্রথমেই দেখে নিন আপনার হার্ড ডিস্কে সি আছে কিনা? এবার সি সেজ করুন। পর্যায়ে বিভিন্ন অপনামসহকারে সি এর বস দেখতে পারেন।

সি প্রোগ্রামিং এবং টেক্সট এদর্শন

সি তে ফল প্রোগ্রামের চেহারা কেমন হবে আর একটি ছোট মনুলা দেখা হল। প্রোগ্রামটি কিভাবে কাজ করল তা পরে বর্ণনা করা হবে। তবে করুন, পর্যায়ে আপনি শিখতে চাইছেন Welcome to Computer Jagat। প্রোগ্রামটি কিভাবে লিখবেন তা দেখানো হল।

```
#include <stdio.h>
main()
{
    printf("Welcome to Computer Jagat");
}
```

প্রোগ্রাম # ১

প্রথমেই একটি ফাংশন দিয়ে প্রোগ্রাম লিখা শুরু হয়েছে। ফাংশনটি এভাবে main()। main ফাংশনে পাশে কোন পেন্স ছাড়াই প্রথম বন্ধনী () রয়েছে। অর্থাৎ, main() ক্রমাগত পর থেকে প্রোগ্রাম তার কাজ শুরু করে। তারপর সফল কিছুই একটি ব্রুকে জ্ঞাপা হয়েছে। ব্রুকে করা হয়েছে দ্বিতীয় বন্ধনী দিয়ে। প্রথমে দ্বিতীয় বন্ধনীর শুরু '।' এবং শেষে দ্বিতীয় বন্ধনীর শেষ '।'।

এছাড়া প্রোগ্রামের শুরুতেই প্রোগ্রামটিকে নির্দেশনা (directive) দেয়া হয়েছে। এই নির্দেশনা মূল প্রোগ্রামের কোন অংশ নয়; এটি প্রোগ্রামকে অনুবাদ করার জন্য কম্পাইলারের কমান্ড হিসেবে ব্যবহৃত হয় আবার প্রধান ব্রুকের বিভিন্ন ফাংশনের উপর ভিত্তি করে এই নির্দেশনা প্রদান করতে হয়। এই প্রোগ্রামে যেমন দেখা হয়েছে stdio.h (standard input output header)। অনেক কম্পাইলার এই নির্দেশনা ছাড়াই প্রোগ্রামকে কম্পাইল করতে পারে। কিন্তু এই নির্দেশনা দেয়ারিই ভাল। প্রতিটি নির্দেশনা তিনটি এক একটি স্বাইমে সিতে হবে। প্রোগ্রামের মূল অংশে ফাঁকা পেন্স বা ফাঁকা লাইন লেখা বা না লেখার জন্য প্রোগ্রামের কোন অসুবিধা হয় না; তবে নির্দেশনাকে উল্লেখ করা থাকলে সন্দেহের কিছু নিয়ম সত্যতে হবে। Welcome to Computer Jagat সেখানি পর্যায়ে প্রদর্শন করা একটি ফাংশন ব্যবহার করা হয়েছে। যেখানে printf ডেফিনি একটি ফাংশন। printf এর সাথে কোটেশন দেয়া স্বাক্ষরটি হচ্ছে স্ট্রিং। এই ফাংশন এবং স্ট্রিংকে একটি কাঠামোতে সাজাতে হবে। printf ফাংশনটি কিন্তু স্ট্রিংকে উল্লেখ করা ফাংশন হিসাবে ব্যবহৃত হচ্ছে আবার এর শেষে সেমিকোলন (;) দেয়া বাধ্যতামূলক। কারণ প্রতিটি সেমিকোলন (;) স্ট্রিংকে শেষ হয়েছে বুঝায়। ১ নং প্রোগ্রামটি রান করলে পর্যায়ে প্রদর্শিত হবে।

```
Welcome to Computer Jagat
অর্থাৎ, সম্পূর্ণ বক্তব্যটি এক লাইনে প্রদর্শিত হয়েছে। আবার প্রোগ্রামটি যদি মীরে মত হয় তবুও কিছু এক লাইনে বক্তব্যটি প্রদর্শিত হবে।
#include <stdio.h>
main()
{
    printf("Welcome to");
    printf("Computer Jagat");
}
```

প্রোগ্রাম # ২

কিন্তু এটিকে দুই লাইনে অর্থাৎ প্রথম লাইনে Welcome to এবং দ্বিতীয় লাইনে Computer Jagat এভাবে প্রদর্শন করতে চাইলে "নতুন লাইন কার্যকর" বসিয়ে প্রোগ্রামটি লিখতে হবে। নতুন লাইন কার্যকর হচ্ছে \n। \n প্রথম লাইনের শেষে বসাতে হবে। ৩ নং প্রোগ্রামটি দেখুন।

```
#include <stdio.h>
main()
{
    printf("Welcome to\n");
    printf("Computer Jagat");
}
```

প্রোগ্রাম # ৩

এখন ৩ নং প্রোগ্রামের আউটপুট হবে Welcome to Computer Jagat \n (নতুন লাইন কার্যকর) সর্বমুখ্যই পরবর্তী লাইন নির্দেশ করে, পরবর্তী আবার একটি লাইন ফাঁকা রাখতে চাইলে আরও একটি \n রাখতে হবে। অর্থাৎ, \n \n দুই লাইন ফাঁকা রেখে পরবর্তী লাইন বসবে। এখানে উল্লেখ্য \n সর্বমুখ্যই printf ফাংশনের ভিতরে দেয়া স্ট্রিংকেই বোঝানো দিতার প্রায়তে হবে।

এবার ৪ নং প্রোগ্রাম # লক্ষ করুন।

```
#include <stdio.h>
main()
{
    printf("Welcome to Computer");
    printf("Jagat");
}
```

প্রোগ্রাম # ৪

৪ নং প্রোগ্রামের আউটপুট হবে Welcome to Computer Jagat এখানে Computer এবং Jagat এর মাঝে একটি পেন্স রয়েছে; প্রথম ট্রিং Welcome to Computer এবং দ্বিতীয় ট্রিং Jagat কে একটি পেন্স দ্বারা সংযুক্ত করা হয়েছে। এই পেন্সটি প্রথম ট্রিং এর শেষে দিয়ে কোটেশন শেষ করা হয়েছে; কাজেই এমনিট ঘটেছে।

২ এবং ৪ নং প্রোগ্রাম দুটোতে সফল ফলাফল বিশেষ কোন অংশেই উল্লেখ না থাকলে ফাঁকা লাইন বা পেন্স টেক্সট এর অংশ হিসেবে প্রদর্শিত হচ্ছে। এখন আপনি আপনার টিকানা সংলগ্ন একটি প্রোগ্রাম লিখুন। মনুলা দেখবেন ৫ নং প্রোগ্রাম দেখুন।

```
#include <stdio.h>
main()
{
    printf("The Molyth Computer Jagat\n");
    printf("146/1 Azimpur Road, Dhaka-1205\n");
    printf("Bangladesh\n");
    printf("Tel: 866746, Fax: 862192\n");
    printf("-----");
}
```

প্রোগ্রাম # ৫

পর্যায়ে এর আউটপুট দেখুন
The Monthly Computer Jagat
146/1 Azimpur Road, Dhaka-1205
Bangladesh
Tel: 866746, Fax: 862192
সংখ্যা প্রদর্শন

আপনি স্বয়ং সি প্রোগ্রাম দিয়ে কোন সংখ্যা প্রদর্শন করতে চান তখন আপনাকে কিছু বিষয়ের উপর নজর রাখতে হবে। যেমন আপনার দেয়া সংখ্যা কি পূর্ণ সংখ্যা (integer) না স্বী দশমিক বা ত্র্যাংশ সংখ্যা (float)। পূর্ণ সংখ্যা বা দশমিক সংখ্যা উভয়ের জন্যই ফাংশন printf ফাংশনে কোটেশন ডিফার ভিতর দুটি কার্যকর ব্যবহার করতে হবে। পূর্ণ সংখ্যার জন্য %d এবং দশমিক সংখ্যার জন্য %f।

কোটেশন চিহ্নের পর কমা (,) দিয়ে আপনার সংখ্যাটি লিখবেন। ৬ নং প্রোগ্রামটি দেখুন।

```
#include <stdio.h>
main()
{
    printf ("%d\n",146);
    printf ("%f\n", 146.25);
```

প্রোগ্রাম : ৬

৬ নং প্রোগ্রামটির আউটপুট হবে

```
146
146.25
```

উল্লেখিত প্রোগ্রামে %d এবং %f কে সাধারণত Conversion specifier বলা হয়। আরও কনভার্সন স্পেসিফায়ার আছে পরে সেগুলোর আলোচনা আসছে। এরা মেমরীতে একটি স্থান দখল করে নেয় এবং কমপাইলারকে নির্দিষ্ট সংখ্যা প্রোগ্রামে লেখার নির্দেশ দেয়। আবার এরা ক্যারেক্টার বা নির্দিষ্ট সংখ্যাকে ঘরমটি করার জন্যও ব্যবহৃত হয়। পরবর্তীতে এ সম্পর্কে আলোচনা করা হবে। এখানে উল্লেখ্য, পূর্ণ সংখ্যা বা দশমিক সংখ্যার জন্য আলাদা স্পেসিফায়ার রয়েছে এদের মধ্যে ভুল করলে সংখ্যা প্রদর্শন করবে না এমনকি টপা পাস্ট। অন্য কিছু প্রদর্শিত হতে পারে। ৭ নং প্রোগ্রামটি দেখুন।

```
#include <stdio.h>
main()
{
    printf ("%f \n", 146);
    printf ("%f \n", 146.25);
```

প্রোগ্রাম : ৭

আউটপুট হবে

```
0.000000
0
```

অর্থাৎ ভুল কিছু প্রদর্শন করছে। এ বাপায়ে ইনপুটের সময় সতর্ক থাকতে হবে। এরকম প্রোগ্রামের সাহায্যেই আমরা গাণিতিক সমস্যার সমাধান করতে পারি। সেক্ষেত্রে printf ক্যারেক্টারি কাল্য করবে। ধরুন, আমরা 146 কে ২ দিয়ে ৩য় অথবা ভাগ করব। ৮ নং প্রোগ্রামটি দেখুন—

```
#include <stdio.h>
main()
{
    printf ("%d \n", 146*2);
    printf ("%f \n", 146.0/2);
```

প্রোগ্রাম : ৮

প্রোগ্রামটির আউটপুট দেখুন

```
292
123.000000
```

অর্থাৎ আমরা সরাসরি printf ফাংশনে গাণিতিক সূত্রগুলো ব্যবহার করে তার ফলাফল পেতে পারি। এ ক্ষেত্রে ফলাফল %d বা %f এর নিম্ন অনুসারেই প্রদর্শিত হবে। তাপনি আপনার পছন্দ মত সংখ্যা দিয়ে বলিভের চর্চা করুন।

ভেরিয়েবল ব্যবহার

ভেরিয়েবল হচ্ছে মেমরীর লোকেশন ট্রিক করে দেয়া। এর জন্য নাম দিতে হয়। প্রোগ্রামার সাধারণত ক্যারেক্টার অর্থাৎ a-z বা A-Z বা 0-9, আন্ডারস্কোর () দিয়ে এই ভেরিয়েবল নির্ধারণ করেন। এক্ষেত্রে নি এর নিজস্ব কিছু নাম সংরক্ষিত আছে। সেগুলো অবশ্যই যান দিয়ে ভেরিয়েবল লিখতে হবে (পরে এদের তালিকা দেব)। যেমন int, main, float ইত্যাদি সি এর কমপাইলারের জন্য সংরক্ষিত আছে। এই ভেরিয়েবল অনুর্ভ ৩২টি ক্যারেক্টার পর্যন্ত ব্যবহার করে রাখা উচিত। কারণ ৩২ ক্যারেক্টার পর্যন্ত ভাগ পর্যন্ত বহন করে পরবর্তীতে নাম। আরেকটি কথা ভেরিয়েবলের জন্য ছোট হাতের বা বড় হাতের যে কোন ক্যারেক্টারই ব্যবহার করা যেতে পারে। একই নাম ছোট হাতের এবং বড় হাতের গিয়ে ভেরিয়েবল হিসেবে ব্যবহার করলে প্রতিটি কিছু আলাদা হিসেবেই ব্যবহৃত হবে। যেমন JAGAT এবং jagat একই নাম কিন্তু ভেরিয়েবল এক নয়। এবার আমরা ভেরিয়েবল বিভাগে ডিক্রয়ার করতে হয় তা দেখব।

ভেরিয়েবল ডিক্রয়ার করণ

ভেরিয়েবল অবশ্যই প্রোগ্রামের প্রথমে ডিক্রয়ার করতে হয়। নিচের উদাহরণটি

```
লক্ষ্য করুন
#include <stdio.h>
main()
{
    int a;
    float b;
    printf ("%d\n",a);
    printf ("%f\n",b);
```

প্রোগ্রাম : ৯

প্রোগ্রামটিতে দুটি ভেরিয়েবল ডিক্রয়ার করা হয়েছে। একটি integer হিসেবে a এবং অপরটি float হিসেবে b। এখানে যদিও ভেরিয়েবল ডিক্রয়ার করা হয়েছে কিন্তু এদের কোন মান দেয়া হয়নি। কাজেই এ প্রোগ্রামটির আউটপুট কিছু আশা করা যায় না। আবার বরং মান বলিয়ে একটি প্রোগ্রাম লিখি তারপর আউটপুট দেখি। পরবর্তী প্রোগ্রাম দেখুন।

```
#include <stdio.h>
main()
{
    int a;
    float b;
    a = 1024;
    b = 512;
    printf ("%d\n", a);
    printf ("%f\n", b);
```

প্রোগ্রাম : ১০

এখানে দুটি ভেরিয়েবল ডিক্রয়ার করা হয়েছে এবং এদের মানও দেয়া হয়েছে। মান দেয়ার ব্যাপারটিকে কপা হয় assign করা। অর্থাৎ 1024 কে ভেরিয়েবল a তে এবং 512 কে b তে assign করা হয়েছে। এবার কিছু আমরা আউটপুট পাবো। প্রোগ্রামটি রান করলে দেখা যাবে :

```
1024
512.000000
```

আবার ভেরিয়েবলের মাধ্যমে অন্য ভেরিয়েবল বা ভেরিয়েবলের এক্সপ্রেশনকে assign করা হয়েছে। প্রোগ্রাম ১১ লক্ষ্য করুন :

```
#include <stdio.h>
main()
{
    int a;
    float b, c;
    a = 1024;
    b = a / 2.0;
    c = b + a
    printf (" Result is %f\n", c);
```

প্রোগ্রাম : ১১

প্রোগ্রামটিতে তিনটি ভেরিয়েবল ডিক্রয়ার করা হয়েছে। তবে এখানে a/2.0 এক্সপ্রেশনটিকে ভেরিয়েবল b তে এবং a ও b এর যোগফলকে ভেরিয়েবল c তে assign করা হয়েছে। এবার যদি প্রোগ্রামটির আউটপুট দেখি অঙ্কলে পাবো।

```
Result is 1536.000000
```

এখানে উল্লেখ্য float ভেরিয়েবলে integer সংখ্যা দিলেও ফলাফল দশমিকে পাওয়া যায়। কিন্তু ভেরিয়েবল হিসেবে float উল্লেখ করা না থাকলে পূর্ণ সংখ্যা বল দিয়ে দশমিকের পরের অংশ লোপে না। ১২ নং প্রোগ্রামটি দেখুন।

```
#include <stdio.h>
main()
{
    int a, b;
    a = 10;
    b = a/4
    printf ("%d\n", b);
```

এক্ষেত্রে আউটপুট পাওয়া যাবে ২। অর্থাৎ দশমিকে ফলাফল আসবে না। কাজেই প্রোগ্রাম লেখার সময় কিছুটা সচেতন হওয়া প্রয়োজন। অন্যথায় সঠিক ফলাফল পাওয়া যাবে না। প্রোগ্রাম কোডিং এর সময় ভুল করলে কমপাইল করার সময় ভাঙ্গি মেসেজ (error message) দেখাবে। অনেক সময় ভাঙ্গি মেসেজ না দেখিয়ে প্রোগ্রাম কমপাইল হয়ে যাবে। সেক্ষেত্রে ফলাফল ভুল আসবে। (সেই)