

সি এর প্রথম পাঠ: ইনপুট হতে আউটপুট

এই টিউটরিয়ালটি তৈরী করা হলো তাদের জন্য যারা DOSBox-এ Turbo C (কোথাও TC নামে পরিচিত) বা বোল্যান্ড সি ইন্সটল করতে ও কনফিগার করতে জানে। কেবল সি প্রোগ্রামিং শিখতে শুরু করেছে বা ভার্শিটিতে কেবল ভর্তি হয়েছে তাদের জন্য কাজে দিবে এই টিউটরিয়াল। এই টিউটরিয়ালটি প্রথমে একবার পড়ে নিন। তারপর শুরু থেকে টিউটরিয়ালটি হাতে কলমে চর্চা করুন।

=====

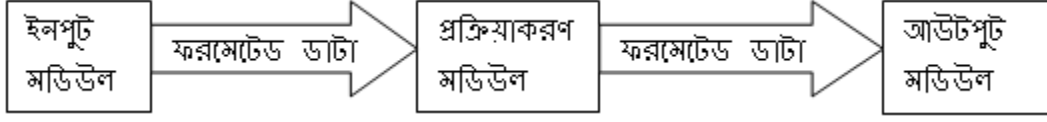
টিউটরিয়াল ১

----- সি এর প্রথম পাঠ: ইনপুট হতে আউটপুট -----

আমাদের সবার মনে আছে কিনা জানি না যে যেদিন (ডিসেম্বর/জানুয়ারী মাসে) স্কুলে পরীক্ষার ফলাফল দিত সেদিন অনেক প্রস্তুতি নিয়ে একটু আগে ভাগেই স্কুলে যেতাম। আমাদের সময় ডিসেম্বর মাসের শুরুতে হত বর্ষ সমাপনী পরীক্ষা। পরীক্ষার খাতা মূল্যায়নের পর প্রস্তুত করা হয় ফলাফল। ফলাফল প্রকাশের দিন প্রধান শিক্ষক বা বিদ্যালয়ের সিনিয়র শিক্ষক সে ফল প্রকাশ করতেন। এখনো হয়ত তাই হয়। প্রধান শিক্ষক এসে প্রথমেই কিছু উপদেশমূলক বাণী দেন। ফলাফলের পাতা উল্টে শুরুতে পড়েন ফলাফলের পরিসংখ্যান। যেমন এ বছর মোট ১২০০ জন পরীক্ষায় অংশগ্রহণ করে। তার মধ্যে ১১০০ জন পাশ করেছে। ব্লা ব্লা ব্লা।

এই পরীক্ষা পদ্ধতিটিকে যদি একটি প্রোগ্রাম হিসাবে ধরে নেই। তাহলে ছাত্ররা যে পরীক্ষার খাতায় লিখে তাহল এর ইনপুট। শিক্ষক কর্তৃক খাতা মূল্যায়ন করার পদ্ধতিটি হল প্রক্রিয়াকরণ। আর প্রধান শিক্ষক কর্তৃক ফলাফল প্রকাশ করা হল প্রোগ্রামটির আউটপুট। কাজেই এই আলোচনায় দেখতে পাই যে একটি প্রোগ্রামের তিনটি প্রধান অংশ রয়েছে। যার

প্রক্রিয়াকরণ অংশ ইনপুটের উপর নির্ভরশীল (যদি ইনপুট অত্যাবশ্যক হয়)। আবার এর আউটপুট অংশ প্রক্রিয়াকরণের উপর নির্ভরশীল। বিষয়টি নিচের চিত্র ২.১-এ দেওয়া হল।



চিত্র ২.১ঃ প্রোগ্রামের বিভিন্ন অংশ।

একটি কম্পিউটার প্রোগ্রাম তার ইনপুট ডাটা ব্যবহারকারীর কাছে চাইতে পারে। আবার ডাটা ইনপুট না নিয়েও কোন ধ্রুবক মান থেকেই একটি প্রোগ্রাম তার কম্পিউটেশন শুরু করে দিতে পারে। অথবা ব্যবহারকারীর নিকট হতে ডাটা না নিয়ে কোন ফাইল বা ক্লাইড হতেও ইনপুট তথ্য নিতে পারে। যাহাই ঘটুক না কেন, আউটপুট থাকবেই। প্রক্রিয়াকৃত ফলাফল আউটপুটে যাবেই।

এবার শুরুতে ধরে নেই যে ব্যবহারকারীর নিকট হতে ডাটা না নিয়েই প্রোগ্রাম তার প্রক্রিয়াকরণ অংশের ডাটা সরাসরি আউটপুটে পাঠাবে। যেমন ধরি যে প্রধান শিক্ষক তার ছাত্রদের উদ্দেশ্যে শুরুতেই বললেন যে, "Hello, dear students! How are you?"। এখন এই বার্তাটুকু প্রোগ্রামের মাধ্যমে আউটপুটে দিতে চাই। তাহলে কি করতে হবে? এই কথাটিকে আউটপুটে (মনিটরে) নিয়ে যাওয়ার জন্য একাধিক ইন্সট্রাকশন দরকার। এই সকল ইন্সট্রাকশন সেট সিপিও হতে বার্তাকে একটি নির্ধারিত পোর্ট দিয়ে মনিটরে পৌঁছে দেওয়ার দায়িত্ব নেয়। শুধু এই কাজটির জন্যই অনেক বড় একটি প্রোগ্রাম দরকার। এরকম নানান কাজের জন্য C কম্পাইলারের নিজস্ব ফাংশন রয়েছে। এ রকম বিবিধ সাধারণ কাজের জন্য C কম্পাইলারের বহু সংখ্যক বিল্ট-ইন ফাংশন রয়েছে। এগুলিকে বলা হয় হেডার/বিল্ট-ইন/লাইব্রেরী ফাংশন। ব্যবহারের ধরন অনুযায়ী এসকল হেডার ফাংশন গুলিকে সমগোত্রে বিভাজন করে আলাদা আলাদা ফাইলে রাখা হয়। এ সকল ফাইলকে বলা হয় হেডার ফাইল। যেমন, স্ট্যান্ডার্ড ইনপুট-আউটপুট সংক্রান্ত `stdio.h`, বিভিন্ন স্ট্যান্ডার্ড লাইব্রেরী সংক্রান্ত হলে `stdlib.h`, ম্যাথ সংক্রান্ত বিষয়ে `math.h`, গ্রাফিক্স নিয়ে হলে `graph.h`, ইত্যাদি। প্রধান শিক্ষকের বার্তাকে প্রদর্শন করার জন্য আমরা `printf` ফাংশনকে ব্যবহার করব। এই `printf` ফাংশনটি ব্যবহারের পূর্বে `stdio.h` হেডার ফাইলকে প্রোগ্রামের শুরুতে ইনক্লোড করতে হবে। এখন এ বিষয়ে প্রোগ্রামটি দেখিঃ

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

```
void main()
```

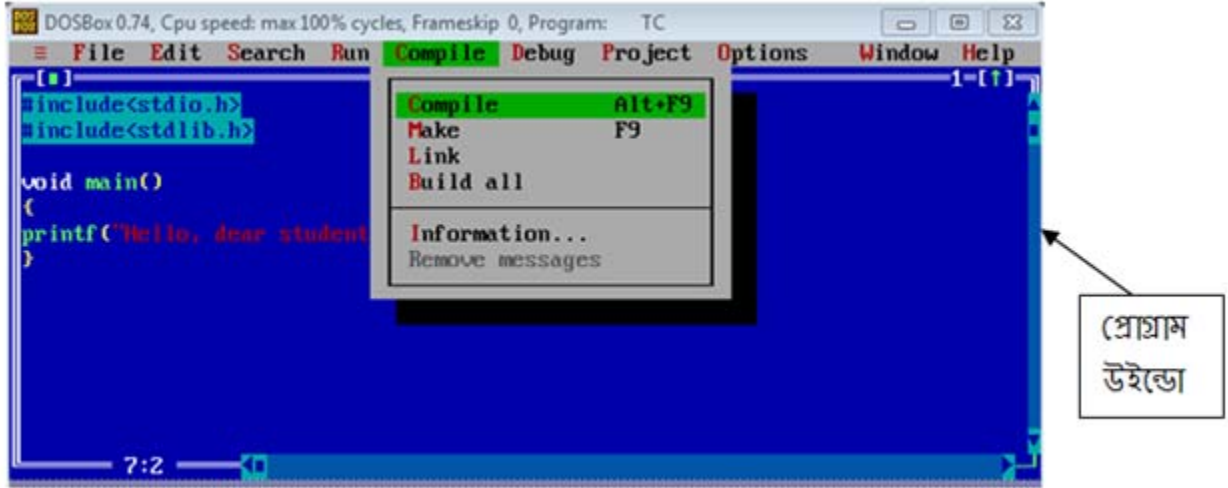
```
{
```

```
printf("Hello, dear students! How are you?");
```

```
}
```

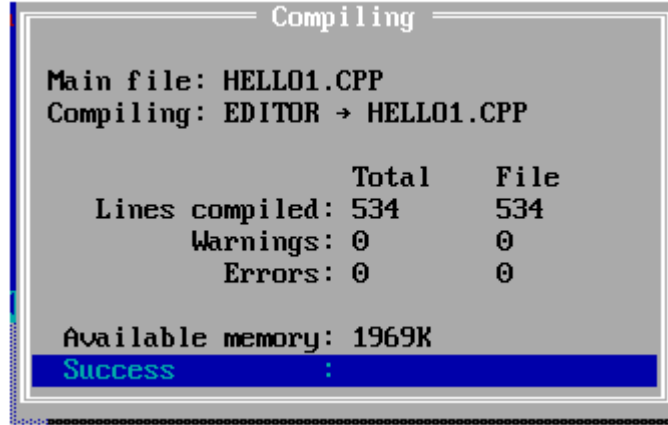
উপরের এই প্রোগ্রামকে টাইপ করার পর একে একটি নামে সেভ করতে হবে যাতে করে পরবর্তীতে তা আবার খুঁজে পাওয়া যায়। এবার ধারাহাসিক ভাবে নিচের বিষয়গুলি অনুসরণ করতে হবে।

১। মেনুবারে কম্পাইল-এ ক্লিক করে কম্পাইল অপশন সিলেক্ট করলে কম্পাইল হয়ে যাবে। কি বোর্ডে Alt চেপে ধরে রেখে F9 চাপলেও একই কাজ হবে। নিচের চিত্র ২.২-এ তা দেখানো হল।

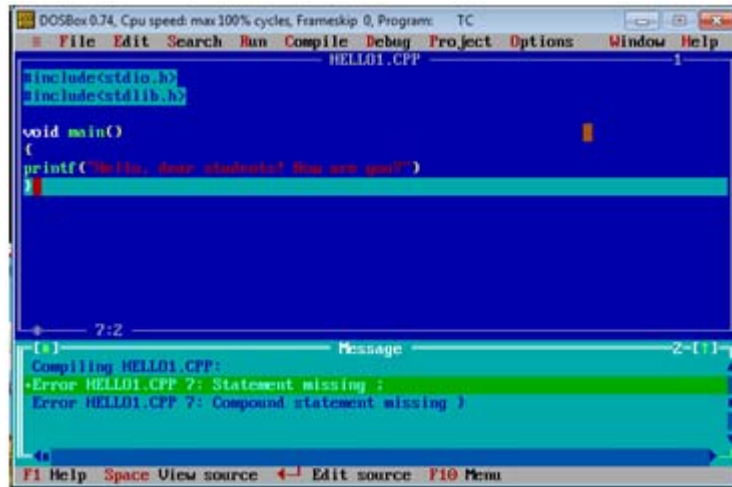


চিত্র ২.২ঃ কম্পাইল করণ।

২। প্রোগ্রাম উইন্ডোর উপর নিচের চিত্র ২.৩-এর ন্যায় একটি আলাদা বক্স আসবে। যদি কোন এরর থাকে তবে এ বক্সে তা সংখ্যা আকারে দেখাবে। যেমন, আমার প্রোগ্রামে ইরর-এর সংখ্যা হল ০। যদি এরর থাকে, তাহলে ইন্টার বাটন চাপলেই নিচের দিকে সকল এরর দেখাবে। আমরা যদি উপরের প্রোগ্রামে printf ফাংশনের শেষে সেমিকোলন না দেই, তাহলে চিত্র ২.৪-এর ন্যায় এরর লিস্ট দেখাবে। সেই তালিকার বার্তা দেখেই বুঝতে হবে এটি কোন ধরনের এরর। সে মোতাবেক ব্যবস্থা নিয়ে আবার কম্পাইল করতে হবে। একাজটি (কম্পাইল করা+ইরর কারেকশন করা) করে যেতে হবে যতক্ষণ না এরর-এর সংখ্যা ০ হয়।

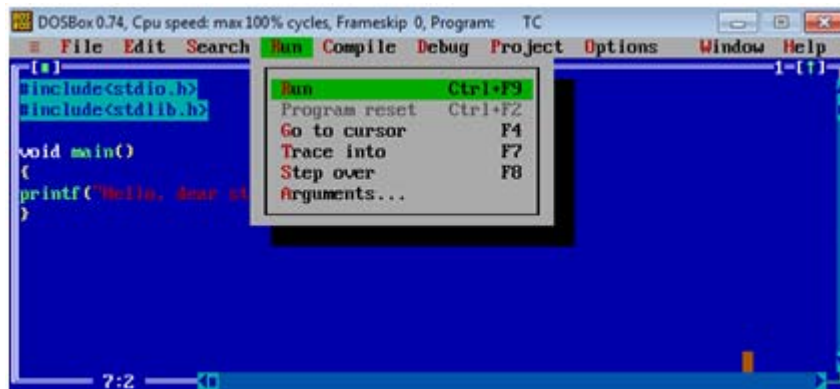


চিত্র ২.৩ঃ কম্পাইল পরিসংখ্যান।



চিত্র ২.৪ঃ ইরর এক্সপ্লোরেশন।

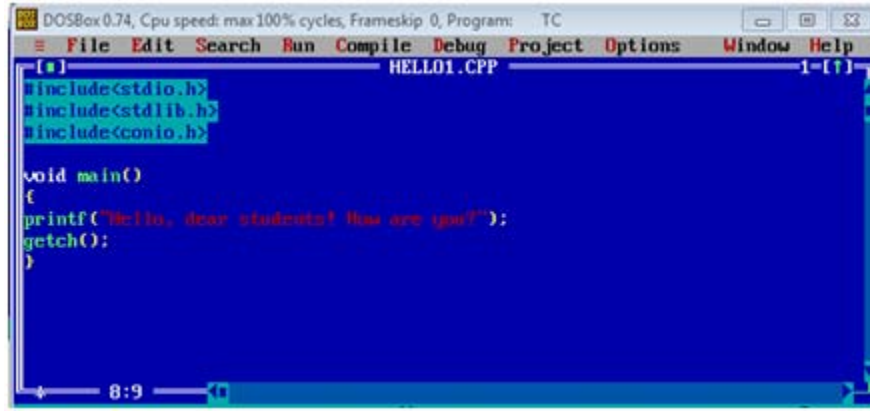
৩। কম্পাইল সাক্ষেস হলে, মেনুবারে রান মেনুতে ক্লিক করতে হবে। তারপর রান অপশনে ক্লিক করলেই প্রোগ্রাম রান হয়ে যাবে। কিবোর্ডে Ctrl চেপে ধরে রেখে F9 চাপলেও রান হয়ে যাবে।



চিত্র ২.৫ঃ প্রোগ্রামকে রান করা।

৪। উপরের প্রোগ্রাম অনুযায়ী আউটপুটে কিছুই দেখবেন না। একটি ক্ল্যাশ দিয়েই আবার প্রোগ্রাম উইন্ডো প্রদর্শিত হবে। ফলাফল দেখতে হলে মেনুবারে উইন্ডো মেনুতে ক্লিক করে আউটপুটে ক্লিক করতে হবে। তাহলেই নিচের দিকে দেখা যাবে যে **Hello, dear students! How are you?!**

৫। আবার যদি রান করার সময়ই সরাসরি আউটপুট দেখতে চান, তাহলে প্রোগ্রামের শেষ } ব্রাকেটের পূর্বে `getch()` হেডার ফাংশনটি ব্যবহার করতে হবে। এই ফাংশনটি ব্যবহার করার জন্য `conio.h` হেডার ফাইলকে শুরুতে ইনক্লুড করতে হবে। তখন প্রোগ্রামটি হবে নিম্নের চিত্র ২.৬ঃ এর মত।

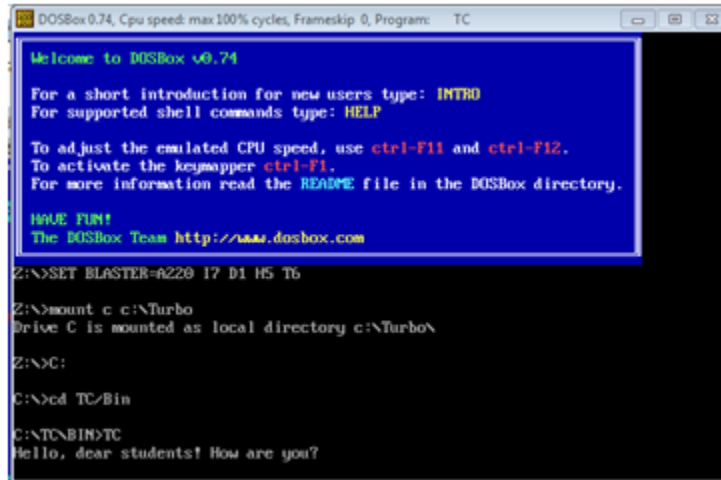


```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
File Edit Search Run Compile Debug Project Options Window Help
HELLO1.CPP 1-[F]
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
printf("Hello, dear students! How are you?!");
getch();
}
```

চিত্র ২.৬ঃ প্রোগ্রাম।

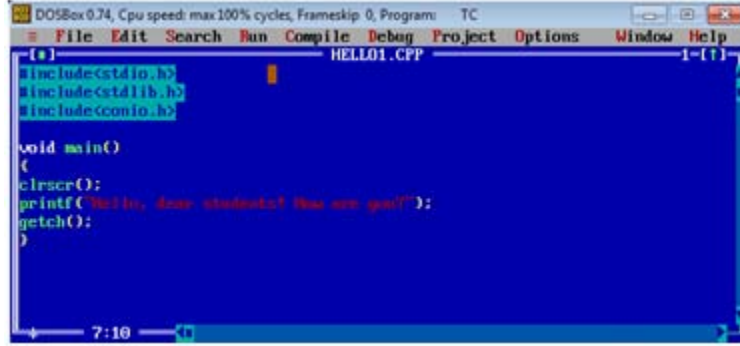
৬। এই প্রোগ্রামটিকে কম্পাইল করে রান করলে নিচের চিত্র ২.৭ দেখাবে। লক্ষ্য করলেই দেখা যাবে যে চিত্রের নিচের দিকে লিখা আছে **Hello, dear students! How are you?!** শেষের দিকে একটি কার্সর ব্লিংকিং করবে। এবার ইন্টার বাটন চাপলেই কার্সর বার প্রোগ্রাম উইন্ডোতে ফেরত আসবে।



```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Welcome to DOSBox v0.74
For a short introduction for new users type: INTRO
For supported shell commands type: HELP
To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>mount c c:\Turbo
Drive C is mounted as local directory c:\Turbo\
Z:\>C:
C:\>cd TC\Bin
C:\TC\BIN>TC
Hello, dear students! How are you?
```

চিত্র ২.৭ঃ আউটপুট উইন্ডো।

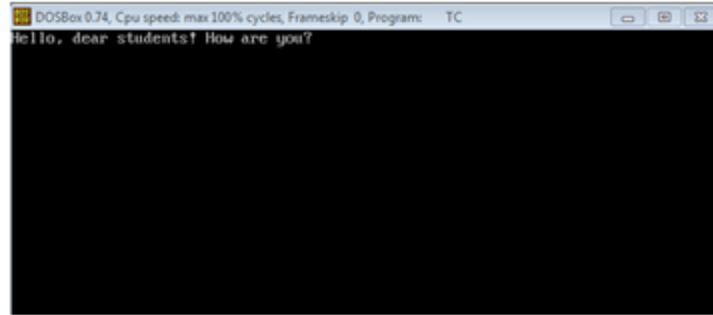
৭। আউটপুট উইন্ডোকে পরিষ্কার করে শুধু প্রোগ্রামের আউটপুট দেখানোর জন্য ক্লিয়ার স্ক্রিন (clrscr) ফাংশন ব্যবহার করতে হয়। clrscr() ফাংশনটি main () এর {ব্রাকেটের পরপরই ব্যবহার করতে হবে। clrscr() ব্যবহার করলে প্রোগ্রাম ও আউটপুট উইন্ডো যথাক্রমে চিত্র ২.৮ এবং ২.৯ এর মত দেখাবে।



```
HELLO1.CPP
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>

void main()
{
clrscr();
printf("Hello, dear students! How are you?");
getch();
}
```

চিত্র: ২.৮ প্রোগ্রাম।



```
Hello, dear students! How are you?
```

চিত্র ২.৯ঃ আউটপুট উইন্ডো।

এবার চিত্র ২.৭ এবং ২.৯ এর তফাৎ দেখুন। এটি clrscr() ফাংশনের কাজ।

উপরের টিউটোরিয়ালের উপর কিছু উদাহরণ দেখি।

উদাহরণ ২.১ঃ আমরা নিচের এই চারটি বাক্যকে মনিটরের আউটপুটে পাঠাতে চাই।

This is a first program. I have successfully executed the program. Everyday, I will spend one hour to be a skilled programmer. I believe that I will be succeeded.

সমাধান ১.১ঃ printf ফাংশনের এর চারটি বাক্যকে এক সাথে প্যারাগ্রাফ আকারে লিখে প্রোগ্রাম রান করলে কাঙ্খিত ফলাফল পাওয়া যাবে। প্রোগ্রামটি নিচে দেখানো হলো।

#include<stdio.h>

```

#include<conio.h>
void main()
{
printf("This is a first program. I have successfully executed the program.
Everyday, I will spend one hour to be a skilled programmer. I believe that I will
be succeeded.");

getch();
}

```

সমাধান ২.২ঃ একাধিক printf ফাংশন ব্যবহার করেও একই কাজ করা যায়। যেমন নিচের প্রোগ্রামে চারটি printf দিয়ে চারটি বাক্যকে একটার পর একটা আউটপুটে পাঠানো হলো। কারণ printf এর কাজই হলো ডবল কোটেশনের ভিতর যা আছে তা আউটপুটে প্রেরণ করা। ফলে উপরের প্রোগ্রামে এক printf ভিতর সমস্ত প্যারা ছিল বলে সবটুকু একসাথে আউটপুটে গেল। আবার নিচের প্রোগ্রামে প্রথম printf এক্সিকিউট হওয়ার পর আউটপুটে যাবে This is a first program. । কিন্তু সি কম্পাইলার লাইন বাই লাইন এক্সিকিউট করে। ফলে দ্বিতীয় printf ও এক্সিকিউট হবে। সে সময় আউটপুটে যাবে This is a first program. I have successfully executed the program. ।

```

#include<stdio.h>
#include<conio.h>
void main()
{
printf("This is a first program. ");
printf("I have successfully executed the program.");
printf(" Everyday, I will spend one hour to be a skilled programmer.");
printf(" I believe that I will be succeeded.");
getch();
}

```

একইভাবে তৃতীয় printf এক্সিকিউট হবার পর আউটপুটে যাবে **This is a first program. I have successfully executed the program. Everyday, I will spend one hour to be a skilled programmer.** সর্বশেষ চতুর্থ printf এক্সিকিউট হলে আউটপুটে যাবে **This is a first program. I have successfully executed the program. Everyday, I will spend one hour to be a skilled programmer. I believe that I will be succeeded.**।

সমাধান ২.৩ঃ

আমরা প্রত্যেকটি printf এর ফলাফল আলাদা আলাদা ভাবে দেখতে পাইনা। কারণ কম্পিউটারের প্রসেসর খুবই ফাস্ট। তাই এত দ্রুত সব এক্সিকিউট হয় যে আমরা এক printf এক্সিকিউট হয়ে অন্য printf এর এক্সিকিউশন শুরু হওয়ার মধ্যে পার্থক্যটুকু ধরতে পারি না। যদি তাও দেখতে চাই তবে একটি printf এক্সিকিউশন হবার পরই প্রসেসরকে হোল্ড করে রাখতে হবে যাতে করে পরের printf এক্সিকিউশন শুরু না করতে পারে। getch() ফাংশন ব্যবহার করে তা করা সম্ভব। নিচের প্রোগ্রামে তাও দেখানো হলো।

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

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

```
void main()
```

```
{
```

```
printf("This is a first program. ");
```

```
getch();
```

```
printf("I have successfully executed the program.");
```

```
getch();
```

```
printf(" Everyday, I will spend one hour to be a skilled programmer.");
```

```
getch();
```

```
printf(" I believe that I will be succeeded.");
```

```
getch();
```

```
}
```

আবার রান করার পর প্রথম আউটপুটে আসবে **This is a first program.** । এবার ইন্টার বাটনে চাপলে আউটপুটে পরের লাইনটি অর্থাৎ **I have successfully executed the program.** আসবে। আর আগে থেকেই **This is a first program.** স্ক্রিনে পাঠানো ছিল। ফলে স্ক্রিনে দেখা যাবে **This is a first program. I have successfully executed the**

program.। একই ভাবে আবার ইন্টার চাপলে আউটপুটে আসবে **This is a first program. I have successfully executed the program. Everyday, I will spend one hour to be a skilled programmer.**। আবার ইন্টার বাটন চাপলে সর্বশেষ **printf** ফাংশনটি এক্সিকিউট হবে। তখন আউটপুটে দেখাবে **This is a first program. I have successfully executed the program. Everyday, I will spend one hour to be a skilled programmer. I believe that I will be succeeded.**।