

# Topic that will be covered

1. Introduction to programming
2. Input-Output
3. Operator
4. Math
5. Control statement -> if,else
6. Switch
7. Loop
8. Series
9. Pattern/pyramic
10. Array (one dimensional)
11. Array (Two dimensional )
12. String
13. Function
14. Recursion
15. File
16. Structure
17. Pointer
18. Advance c

# 1. Introduction to programming

- 5.1 কিভাবে সি প্রোগ্রামিং পড়বেন / পড়াবেন ?
- 5.2 অনুবাদক প্রোগ্রাম - কম্পাইলার, ইন্টারপ্রেটার, অ্যাসেম্বলার
- 5.3 অ্যালগরিদম
- 5.4 ক্লোচার্ট ও সুড়েকোড
- 5.5 প্রোগ্রাম তৈরির ধাপসমূহ ও ডিবাগিং
- 5.6 সি প্রোগ্রামিং পরিচিতি
- 5.7 কিভাবে Code::Blocks IDE সেটআপ করতে হয়

# আজকের বিষয়বস্তু

2

- প্রোগ্রাম কি ? প্রোগ্রামিং ভাষা কি ?
- অনুবাদক প্রোগ্রাম কি ? অনুবাদক প্রোগ্রাম এর প্রকারভেদ আলোচনা কর।
- অ্যাসেম্বলার কি ? এর সুবিধা ও অসুবিধা লিখ।
- কোন কোন প্রোগ্রামিং ভাষাকে কম্পাইল বা ইন্টারপ্রেট করা হয় ?
- কম্পাইলার কি? এটি কিভাবে কাজ করে ?
- ইন্টারপ্রেটার কি? এটি কিভাবে কাজ করে ?
- কম্পাইলার ও ইন্টারপ্রেটার এর পার্থক্য লিখ।

# প্রশ্ন ১. প্রোগ্রাম কি ? প্রোগ্রামিং ভাষা কি ?

3

প্রোগ্রাম- কোন একটি সমস্যা সমাধানের জন্য কম্পিউটারের ভাষায় (0,1) লিখিত নির্দেশের সমষ্টিকে প্রোগ্রাম বলা হয়।



প্রোগ্রামিং ভাষা – প্রোগ্রাম রচনার জন্য বিভিন্ন শব্দ, বর্ণ, অঙ্ক, সংক্ষেত এবং এইগুলো বিন্যাসের নিয়ম মিলিয়ে তৈরি করা হয় প্রোগ্রামিং ভাষা।

যেমন – C, C++, Assembly language, Java ইত্যাদি।

## C programming language

# Translator program (অনুবাদক প্রোগ্রাম)

যে প্রোগ্রাম সোর্স কোডকে অবজেক্ট কোডে  
রূপান্তরিত করে তাকে অনুশীলক প্রোগ্রাম বলে।

ram (অনুবাদক প্রোগ্রাম)

```
Start here step1_002.c ×
1 void doSomething()
2 {
3     __asm__ ("nop");
4 }
5
6 int main()
7 {
8     doSomething();
9     return 0;
10}
11
```

[www.c4learn.com](http://www.c4learn.com)

উচ্চতর ভাষায় লিখিত প্রোগ্রামকে  
সোর্স কোড (source code) বলে।



মেশিন ভাষার কোডকে অবজেক্ট কোড (Object code) বলে

## প্রশ্ন ২. অনুবাদক প্রোগ্রাম কি ? অনুবাদক এর প্রকারভেদ আলোচনা কর।

5

**অনুবাদক প্রোগ্রাম** - যে প্রোগ্রাম সোর্স কোডকে অবজেক্ট কোডে রূপান্তরিত করে তাকে অনুবাদক প্রোগ্রাম বলে।

### অনুবাদক প্রোগ্রাম

ক)  
অ্যাসেম্বলার

খ)  
কম্পাইলার

গ)  
ইন্টারপ্রেটার

# অ্যাসেম্বলার

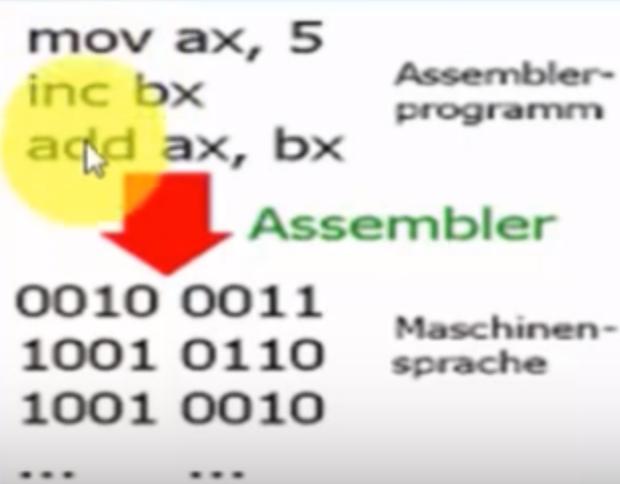
6

Assembly  
Language  
Program

Assembler

Machine  
language

- যে অনুবাদক অ্যাসেম্বলি ভাষায় লিখিত প্রোগ্রামকে মেশিন ভাষায় রূপান্তর করে তাকে অ্যাসেম্বলার বলে।



# অ্যাসেম্বলার এর সুবিধা ও অসুবিধা

7

- অ্যাসেম্বলার এর সুবিধা-
  - ❖ নেমনিক কোডকে মেশিন ভাষায় রূপান্তর করে।
  - ❖ প্রোগ্রামে কোন ভুল থাকলে Error Message প্রদর্শন করে।
  - ❖ প্রোগ্রামের ভুল সংশোধন করা সহজ।
  - ❖ ডিবাগিং ও টেস্টিং এর কাজ দ্রুত গতি সম্পন্ন।
  - ❖ মেমোরিতে কম জায়গা নেয়।
- অ্যাসেম্বলার এর অসুবিধা-
  - ❖ প্রোগ্রাম নির্বাহের জন্য অনেক বেশি সময় লাগে, কম্পাইলার এর তুলনায় প্রায় ৫ থেকে ২৫ গুণ সময় বেশি লাগে।

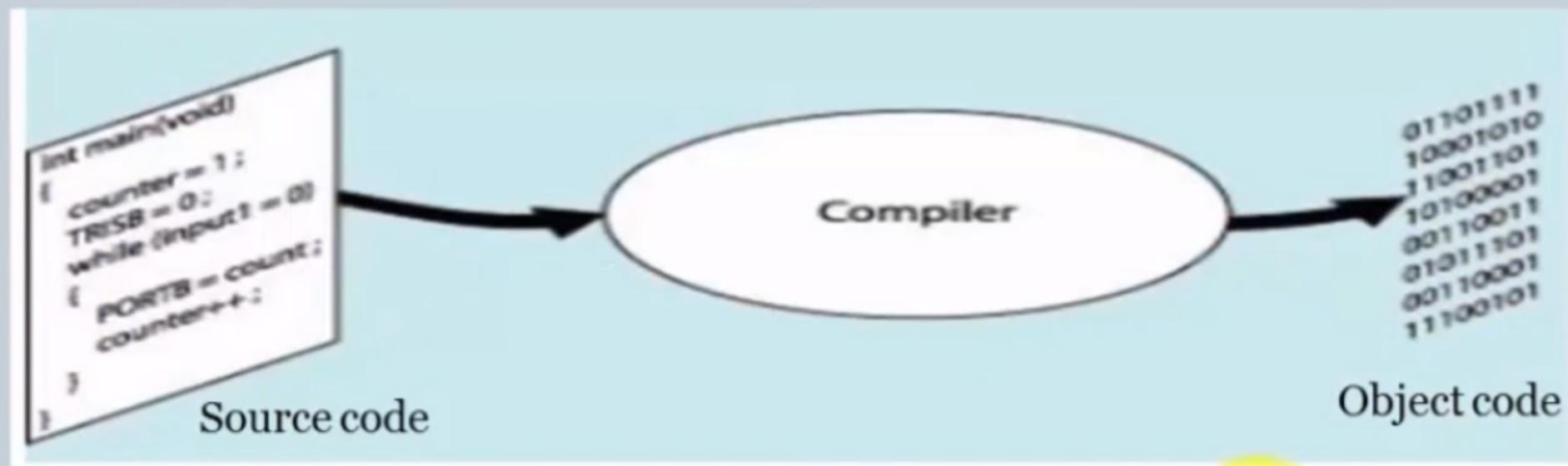
# জেনে রাখা ভাল

8

- যেসব প্রোগ্রামিং ভাষাকে কম্পাইল করা হয় –  
C, C++, Objective-C, C#, Pascal, COBOL, ADA,  
Visual Basic, Smalltalk, Scheme ইত্যাদি।
- যেসব প্রোগ্রামিং ভাষাকে ইন্টারপ্রেট করা হয় –  
BASIC , php, Python, Perl, Ruby, Javascript  
ইত্যাদি।

## কম্পাইলার

9

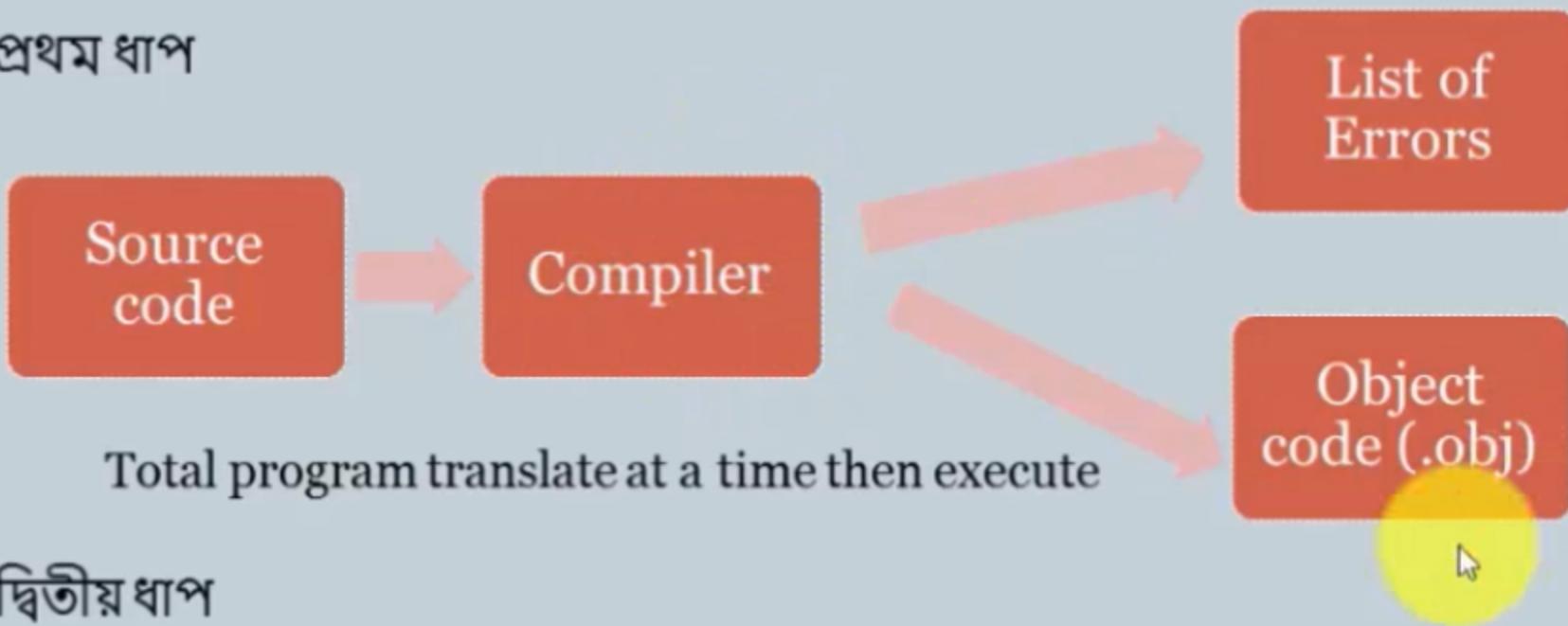


যে অনুবাদক প্রোগ্রাম সোর্স কোডকে অবজেক্ট কোডে  
রূপান্তরিত করে তাকে কম্পাইলার বলে।

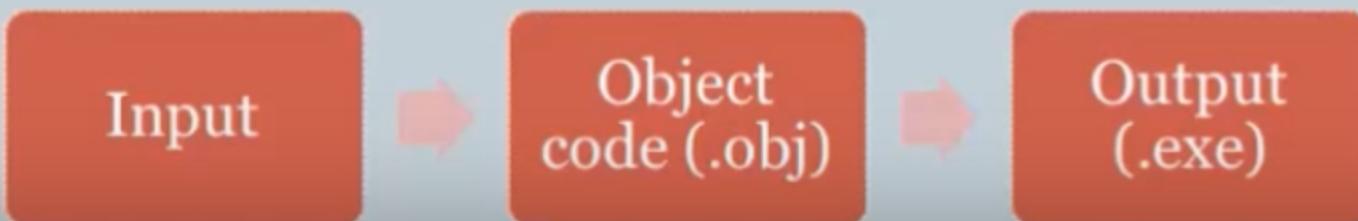
# কম্পাইলারের কাজের ধাপসমূহ

11

## প্রথম ধাপ

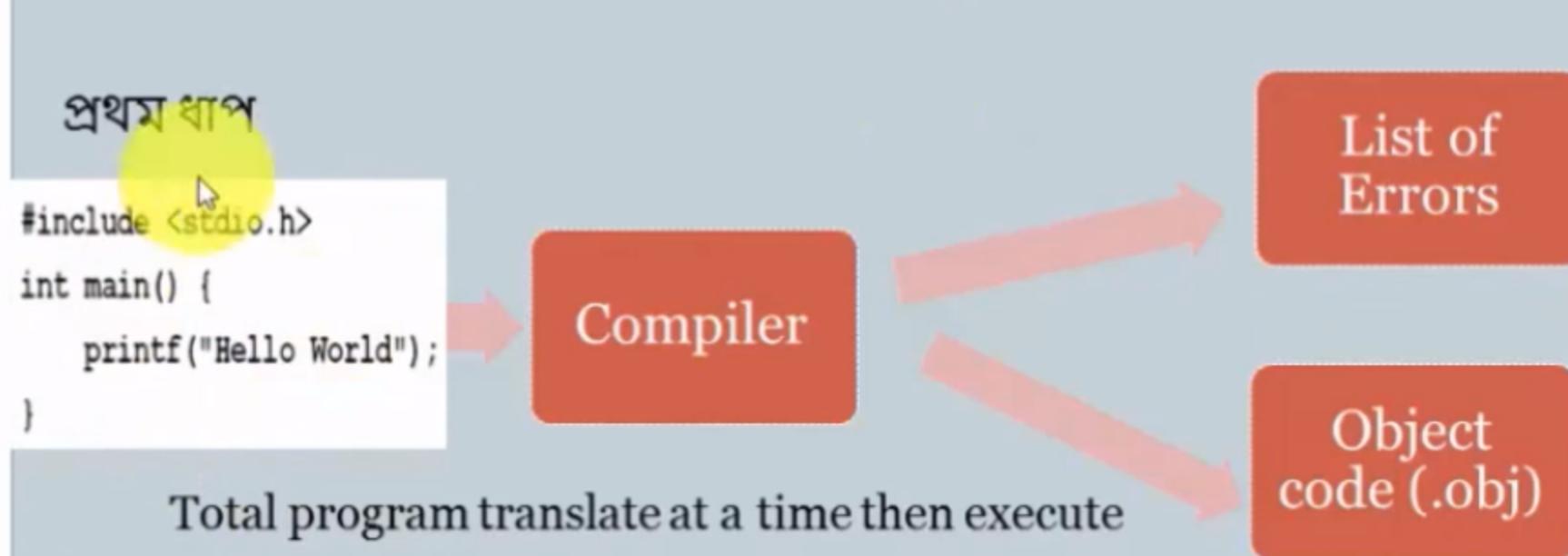
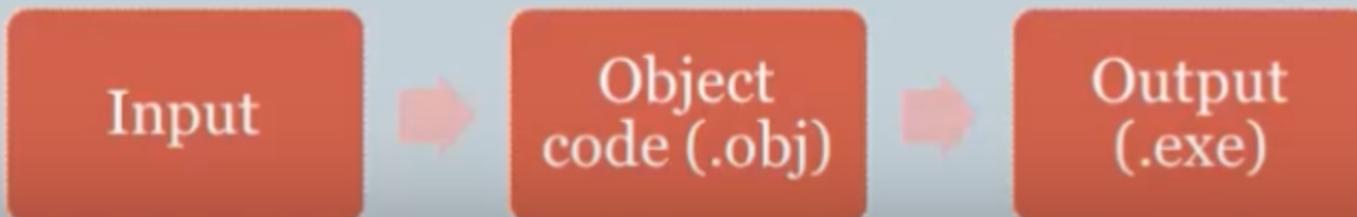


## দ্বিতীয় ধাপ



# কম্পাইলারের কাজের ধাপসমূহ

11

**দ্বিতীয় ধাপ**

# ইন্টারপ্রেটারের কাজের ধাপ

12



Line by Line translate then execute

# ইন্টারপ্রেটারের কাজের ধাপ

12

```
10 REM A simple BASIC program  
20 PRINT "Hello World!"  
30 GOTO 10
```

Interpreter

Output  
(.exe)

Input

Line by Line translate then execute

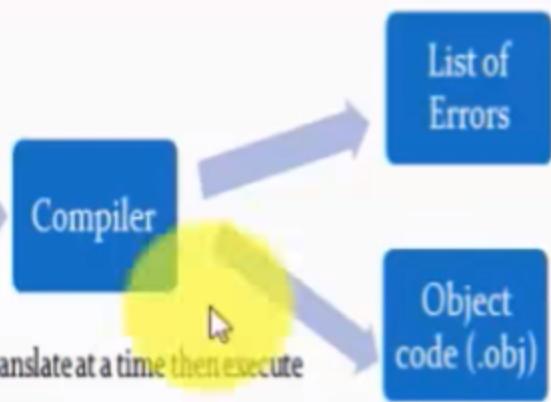
# কম্পাইলার VS ইন্টারপ্রেটার এর কাজের ধাপ

13

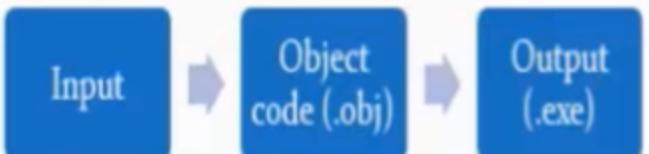
## কম্পাইলারের কাজের ধাপসমূহ

### প্রথম ধাপ

```
#include <stdio.h>
int main() {
    printf("Hello World");
}
```



### বিটীয় ধাপ



## ইন্টারপ্রেটারের কাজের ধাপ

```
10 REM A simple BASIC program
20 PRINT "Hello World"
30 GOTO 10
```



Line by Line translate then execute

# কম্পাইলার ও ইন্টারপ্রেটার এর পার্থক্য লিখ।

14

কম্পাইলার	ইন্টারপ্রেটার
১. জন্মপূর্ণ প্রোগ্রামটিকে একসাথে অনুবাদ করে।	১. এক লাইন এক লাইন করে অনুবাদ করে।
২. প্রোগ্রামের সব ভুল একসাথে প্রদর্শন করে।	২. এক লাইন করে ভুল প্রদর্শন করে।
৩. কম্পাইলার দ্রুত কাজ করে।	৩. ইন্টারপ্রেটার ধীরে কাজ করে।
৪. প্রোগ্রাম একবার কম্পাইল করার পর পরবর্তীতে আর কম্পাইল করার প্রয়োজন হয়না।	৪. প্রতিটি কাজের পূর্বে অনুবাদ করার প্রয়োজন হয়।
৫. বড় ধরণের কম্পিউটারে বেশি ব্যাবহার হয়।	৫. মাইক্রো কম্পিউটারে বেশি ব্যাবহার হয়।

# অ্যালগরিদম

2

## প্রশ্ন ১ : অ্যালগরিদম কি ? অ্যালগরিদম এর নিয়ম ও বৈশিষ্ট্য লিখ এবং সুবিধা লিখ ।

**অ্যালগরিদম :** যে পদ্ধতিতে কোন একটি সমস্যাকে ধাপে ধাপে সমাধান করা হয় তাকে অ্যালগরিদম বলে। অ্যালগরিদম অর্থ ধাপে ধাপে সমস্যা সমাধান ।

**উদাহরণ :** তিনটি সংখ্যার যোগফল নির্ণয়ের অ্যালগরিদম লিখ ।

ধাপ - ১ : শুরু ।

ধাপ - ২ : তিনটি সংখ্যা গ্রহণ করা ।

ধাপ - ৩ : সংখ্যা তিনটি যোগ করা ।

ধাপ - ৪ : যোগফল প্রদর্শন (print) করা ।

ধাপ - ৫ : শেষ ।

### অ্যালগরিদম এর নিয়ম ও বৈশিষ্ট্য :

- I. অ্যালগরিদম সহজবোধ্য হতে হবে ।
- II. প্রত্যেকটি ধাপ স্পষ্ট হতে হবে যাতে যেকোনো প্রোগ্রামার সহজেই বুঝতে পারে ।
- III. ধাপে ধাপে সমস্যার সমাধান করতে হবে ।
- IV. একে ব্যাপকভাবে প্রয়োগ করার উপযোগী হতে হবে ।



### অ্যালগরিদম এর সুবিধা :

- I. সহজে প্রোগ্রামের উদ্দেশ্য বুঝতে সাহায্য করে ।
- II. প্রোগ্রামের ভুল নির্ণয়ে সাহায্য করে ।
- III. প্রোগ্রাম পরিবর্তন ও পরিবর্ধনে সাহায্য করে ।
- IV. সহজে জটিল প্রোগ্রাম লিখতে সাহায্য করে ।

## অ্যালগরিদম এর আরেকটি উদাহরণ

3

প্রশ্ন ২ : একটি ত্রিভুজের ক্ষেত্রফল নির্ণয়ের অ্যালগরিদম লিখ ।

ধাপ - ১ : শুরু ।

ধাপ - ২ : ত্রিভুজের ভূমি ও উচ্চতা গ্রহণ ।

ধাপ - ৩ : ক্ষেত্রফল =  $1/2 \times$  ভূমি  $\times$  উচ্চতা সূত্র ব্যাবহার করে  
ত্রিভুজের ক্ষেত্রফল নির্ণয় করা ।

ধাপ - ৪ : ত্রিভুজের ক্ষেত্রফল প্রদর্শন করা ।

ধাপ - ৫ : শেষ ।

# অ্যালগরিদম এর আরেকটি উদাহরণ

4

প্রশ্ন ৩ : ২ টি পূর্ণ সংখ্যার মধ্যে বড় সংখ্যাটি বের করার জন্য অ্যালগরিদম লিখ।

ধাপ - ১ : শুরু ।

ধাপ - ২ : সংখ্যা ২ টি গ্রহণ ।

ধাপ - ৩ : ১ম সংখ্যাটি কি ২য় সংখ্যাটি হতে বড়?

    ক) হ্যাঁ, ফলাফল ছাপি ১ম সংখ্যাটি বড়।

    খ) না, ফলাফল ছাপি ২য় সংখ্যাটি বড়।

ধাপ - ৫ : শেষ ।

# ফ্লোচার্ট

5

প্রশ্ন ৪ : ফ্লোচার্ট বা প্রবাহচিত্র কি ? ফ্লোচার্ট কত প্রকার কি ? বর্ণনা কর।

- **ফ্লোচার্ট**: যে পদ্ধতিতে প্রোগ্রামকে চিত্রের বা প্রতীকের সাহায্যে প্রকাশ করা হয় তাকে ফ্লোচার্ট বলে।

**উদাহরণ :** তিনটি সংখ্যার যোগফল নির্ণয়ের অ্যালগরিদম ও ফ্লোচার্ট লিখ।

অ্যালগরিদম	ফ্লোচার্ট
ধাপ - ১ : শুরু।	<pre> graph TD     START([START]) --&gt; READ[/READ A,B,C/]     READ --&gt; SUM[SUM = A + B + C]     SUM --&gt; PRINT[/PRINT SUM/]     PRINT --&gt; END([END])     </pre>
ধাপ - ২ : তিনটি সংখ্যা গ্রহণ করা।	
ধাপ - ৩ : সংখ্যা <b>তিনটি</b> যোগ করা।	
ধাপ - ৪ : যোগফল প্রদর্শন করা।	
ধাপ - ৫ : শেষ।	

# ফ্লোচার্ট এর প্রকারভেদ

6

- ফ্লোচার্ট মূলত ২ প্রকার।

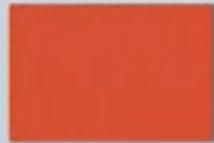


- **সিস্টেম ফ্লোচার্ট**: যে ফ্লোচার্টে একটি সিস্টেমের ডেটার প্রবাহ, প্রক্রিয়াকরণ, সংরক্ষন ও ফলাফল প্রদর্শনের দিক চিহ্নিত হয় তাকে সিস্টেম ফ্লোচার্ট বলে।
- **প্রোগ্রাম ফ্লোচার্ট**: প্রোগ্রাম রচনার ক্ষেত্রে যে ফ্লোচার্ট তৈরি করা হয় তাকে প্রোগ্রাম ফ্লোচার্ট বলে।



# সিস্টেম ফ্লোচার্ট

7



প্রক্রিয়াকরণ



ডকুমেন্ট



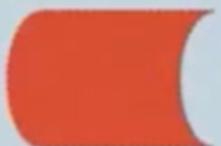
ম্যাগনেটিক টেপ



ম্যাগনেটিক ডিস্ক



কার্ড



অন-লাইন স্মৃতি



ম্যানুয়াল ইনপুট সেটিং বা সাজানো



# প্রোগ্রাম ফ্লোচার্ট

×
8

## প্রশ্ন ৫ : প্রোগ্রাম ফ্লোচার্ট এর প্রতীকগুলো অর্থসহ লিখ।

প্রতীক	অর্থ
	টার্মিনাল প্রতীক, এটি ডিস্বিউ আকৃতির। এটি প্রোগ্রামের শুরু বা শেষ করতে ব্যবহার হয়। (START / END)
	গ্রহণ / নির্গমন প্রতীক, এটি সামান্তরিক আকৃতির। এটি প্রোগ্রামে ইনপুট ও আউটপুট এর ক্ষেত্রে ব্যবহার হয়। (INPUT / OUTPUT)
	প্রক্রিয়াকরণ প্রতীক, এটি আয়তাকার আকৃতির। এটি প্রক্রিয়াকরণের জন্য ব্যবহার হয়। (গাণিতিক হিসাব-নিকাশ এবং বিভিন্ন ধরনের প্রক্রিয়াকরণ)
	সিদ্ধান্ত প্রতীক, এটি হীরক (diamond) আকৃতির। এটি সিদ্ধান্ত গ্রহণের জন্য ব্যবহার হয়, যার দুটি সম্ভাব্য মান থাকে – হ্যাঁ বা না।
	তীরচিহ্ন বিশিষ্ট সরলরেখা প্রোগ্রাম প্রবাহের দিক নির্দেশ করে।
	সংযোগ প্রতীক, যখন একটি বড় ফ্লোচার্ট এক পৃষ্ঠায় জায়গা হয় না তখন এই প্রতীক ব্যবহার করে বাকি অংশ অপর পৃষ্ঠায় আঁকা হয়।

# ফ্লোচার্ট অংকনের নিয়মাবলি

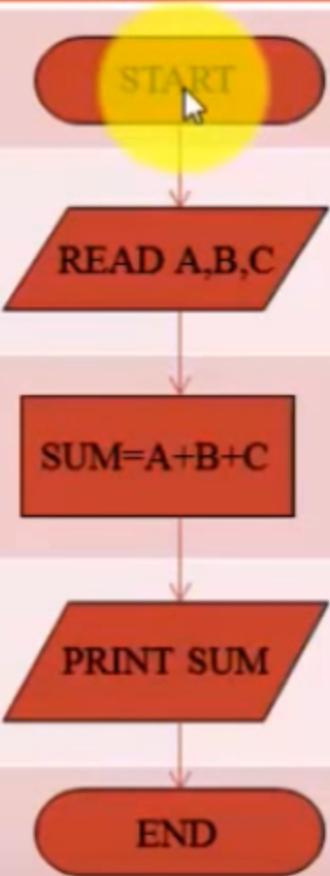
৭

## প্রশ্ন ৬ : ফ্লোচার্ট অংকনের নিয়মাবলী লিখ।

- I. নির্দিষ্ট কিছু প্রতীক ব্যবহার করে ফ্লোচার্ট অংকন করতে হবে।
- II. ফ্লোচার্টের প্রবাহ উপর থেকে নিচের দিকে বা বাম হতে ডান দিকে তীর চিহ্ন দিয়ে দেখাতে হবে।
- III. প্রতিটি প্রতীকের ভিতরের লিখা সহজবোধ্য হতে হবে।
- IV. অতিরিক্ত প্রতিকের ব্যবহার পরিহার করতে হবে।
- V. নির্দিষ্ট কোন প্রোগ্রামিং ভাষা ব্যবহার করা উচিত নয়।

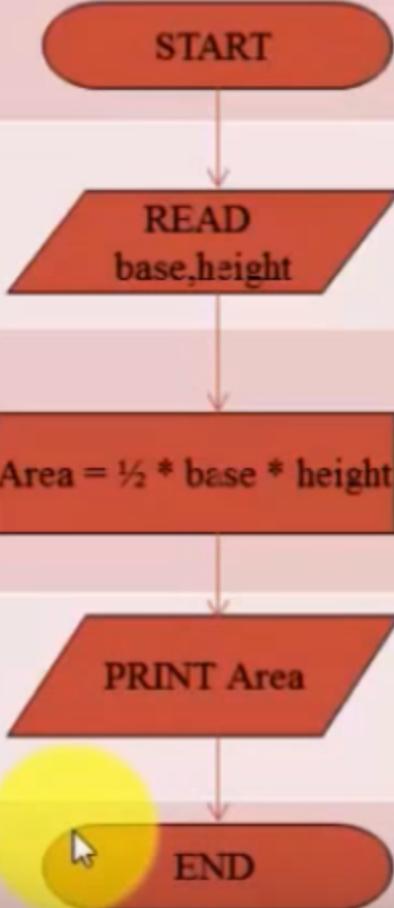
## প্রশ্ন ৭ : তিনটি সংখ্যার যোগফল নির্ণয়ের অ্যালগরিদম ও ফ্লোচার্ট লিখ ।

10

অ্যালগরিদম	ফ্লোচার্ট
ধাপ - ১ : শুরু ।	
ধাপ - ২ : তিনটি সংখ্যা গ্রহণ করা ।	
ধাপ - ৩ : সংখ্যা তিনটি যোগ করা ।	
ধাপ - ৪ : যোগফল প্রদর্শন করা ।	
ধাপ - ৫ : শেষ ।	

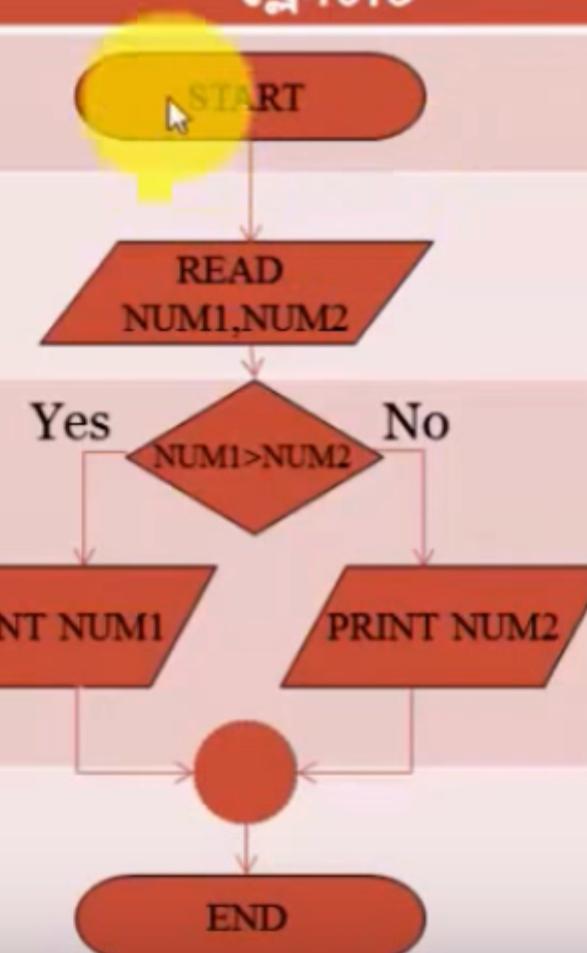
## প্রশ্ন ৮ : একটি ত্রিভুজের ক্ষেত্রফল নির্ণয়ের অ্যালগরিদম ও ফ্লোচার্ট লিখ ।

11

অ্যালগরিদম	ফ্লোচার্ট
ধাপ - ১ : শুরু ।	
ধাপ - ২ : ত্রিভুজের ভূমি ও উচ্চতা গ্রহণ ।	
ধাপ - ৩ : $\text{ক্ষেত্রফল} = 1/2 \times \text{ভূমি} \times \text{উচ্চতা}$ সূত্র ব্যাবহার করে ত্রিভুজের ক্ষেত্রফল নির্ণয় করা	
ধাপ - ৪ : ত্রিভুজের ক্ষেত্রফল প্রদর্শন করা ।	
ধাপ - ৫ : শেষ ।	

## প্রশ্ন ৯ : ২ টি পূর্ণ সংখ্যার মধ্যে বড় সংখ্যাটি বের করার জন্য অ্যালগরিদম ও ফ্লোচার্ট লিখ।

12

অ্যালগরিদম	ফ্লোচার্ট
ধাপ - ১ : শুরু।	
ধাপ - ২ : সংখ্যা ২ টি গ্রহণ।	
ধাপ - ৩ :	
১ম সংখ্যাটি কি ২য় সংখ্যাটি হতে বড়? ক) হ্যাঁ, ফলাফল ছাপি ১ম সংখ্যাটি বড়। খ) না, ফলাফল ছাপি ২য় সংখ্যাটি বড়।	
ধাপ - ৪ : শেষ।	

## অ্যালগরিদম

## ফ্লোচার্ট

১. যে পদ্ধতিতে কোন একটি সমস্যাকে ধাপে ধাপে সমাধান করা হয় তাকে অ্যালগরিদম বলে।

২. এটি বর্ণনা নির্ভর।

৩. প্রোগ্রাম বুঝতে সময় বেশী লাগে।

৪. প্রোগ্রাম প্রবাহের দিক সহজে বুঝা যায়না।

৫. প্রোগ্রামের ভুল-ত্রুটি দূর করা কঠিন।

১. সমস্যা সমাধানের ধাপসমূহকে বিভিন্ন সাংকেতিক চিহ্ন ব্যাবহার করে চিত্রের সাহায্যে প্রকাশ করাকে ফ্লোচার্ট বা প্রবাহচিত্র বলে।

২. এটি চিত্র নির্ভর।

৩. প্রোগ্রাম বুঝতে সময় কম লাগে।

৪. প্রোগ্রাম প্রবাহের দিক সহজে বুঝা যায়।

৫. প্রোগ্রামের ভুল-ত্রুটি দূর করা সহজ।

# অনুশীলন

14

1. ফারেনহাইট তাপমাত্রাকে সেলসিয়াস তাপমাত্রায় রূপান্তরের অ্যালগরিদম ও ফ্লোচার্ট লিখ।
2. বৃত্তের ক্ষেত্রফল নির্ণয়ের অ্যালগরিদম ও ফ্লোচার্ট লিখ।
3. একটি ধনাত্মক পূর্ণসংখ্যার ফ্যাক্টরিয়ল নির্ণয়ের জন্য অ্যালগরিদম ও ফ্লোচার্ট লিখ।
4. ২ টি পূর্ণসংখ্যার মধ্যে ছোট / বড় সংখ্যাটি বের করার জন্য অ্যালগরিদম ও ফ্লোচার্ট লিখ।
5. ৩ টি পূর্ণসংখ্যার মধ্যে ছোট / বড় সংখ্যাটি বের করার জন্য অ্যালগরিদম ও ফ্লোচার্ট লিখ।

# সুড়োকোড (Pseudo code)

15

প্রশ্ন ১১ : সুড়োকোড কি ?

উত্তর : সুড়ো (Pseudo) একটি গ্রিক শব্দ। সুড়ো শব্দের অর্থ হচ্ছে ছদ্ম বা কৃত্রিম। এটি দেখতে অনেকটা প্রোগ্রামিং কোডের মত মনে হবে, কিন্তু আসলে তা নয়। প্রোগ্রামাররা অনেক সময় প্রোগ্রামের জন্য যে কোড লেখা হবে তার একটি খসড়া টেক্সট ভার্সন তৈরি করেন, আর তাকেই সুড়োকোড বলে।

**উদাহরণ :** তিনটি সংখ্যার যোগফল নির্ণয়ের সুড়োকোড লিখ।

```

INPUT NUMBER1
INPUT NUMBER2
INPUT NUMBER3
TOTAL=NUMBER1+NUMBER2+NUMBER3
PRINT TOTAL
STOP
    
```



## আজকে আমরা যা যা শিখব

×

2

১. প্রোগ্রাম তৈরির ধাপসমূহ আলোচনা কর।
২. ডিবাগিং সম্পর্কে আলোচনা কর।

# প্রোগ্রাম তেরির ধাপ

3

- প্রোগ্রামের সাহায্যে কোন সমস্যা সমাধান করা হয়। আর সমস্যা সমাধানের জন্য কতকগুলো ধাপ অনুসরণ করতে হয়। এই ধাপ সমূহকে প্রোগ্রাম তেরির ধাপ বলা হয়।

# প্রোগ্রাম তৈরির ধাপসমূহ

4

১. সমস্যা নির্দিষ্টকরণ



২. সমস্যা বিশ্লেষণ



৩. প্রোগ্রাম ডিজাইন



৪. প্রোগ্রাম উন্নয়ন



৫. প্রোগ্রাম বাস্তবায়ন



৬. প্রোগ্রাম রক্ষণাবেক্ষণ

- ❖ সমস্যা চিহ্নিকরণ
- ❖ সমস্যা সম্পর্কিতত্ব সংগ্রহ

# প্রোগ্রাম তৈরির ধাপসমূহ

5

১. সমস্যা নির্দিষ্টকরণ

২. সমস্যা বিশ্লেষণ

৩. প্রোগ্রাম ডিজাইন

৪. প্রোগ্রাম উন্নয়ন

৫. প্রোগ্রাম বাস্তবায়ন

৬. প্রোগ্রাম রক্ষণাবেক্ষণ

- ❖ ইনপুট শনাক্তকরণ
- ❖ আউটপুট শনাক্তকরণ
- ❖ কম্পিউটারের মেমোরি নির্ধারণ করা

ভূমি ও উচ্চতা গ্রহণ করতে হবে।  
ক্ষেত্রফল প্রদর্শন করতে হবে।

# প্রোগ্রাম তৈরির ধাপসমূহ

6

১. সমস্যা নির্দিষ্টকরণ



২. সমস্যা বিশ্লেষণ



৩. প্রোগ্রাম ডিজাইন



৪. প্রোগ্রাম উন্নয়ন



৫. প্রোগ্রাম বাস্তবায়ন



৬. প্রোগ্রাম রক্ষণাবেক্ষণ

- ❖ অ্যালগরিদম
- ❖ ফ্রেমাচার্ট

# প্রোগ্রাম তৈরির ধাপসমূহ

6

১. সমস্যা নির্দিষ্টকরণ



২. সমস্যা বিশ্লেষণ



৩. প্রোগ্রাম ডিজাইন



৪. প্রোগ্রাম উন্নয়ন



৫. প্রোগ্রাম বাস্তবায়ন



৬. প্রোগ্রাম রক্ষণাবেক্ষণ

পদ্ধতি ২: একটি ত্রিভুজের ক্ষেত্রফল নির্ণয়ের অ্যালগরিদম লিখ।

ধাপ - ১: শুরু।

ধাপ - ২: ত্রিভুজের ভূমি ও উচ্চতা গ্রহণ।

ধাপ - ৩:  $\text{ক্ষেত্রফল} = 1/2 \times \text{ভূমি} \times \text{উচ্চতা}$  সূত্র ব্যবহার করে  
ত্রিভুজের ক্ষেত্রফল নির্ণয় করা।

ধাপ - ৪: ত্রিভুজের ক্ষেত্রফল প্রদর্শন করা।

ধাপ - ৫: শেষ।

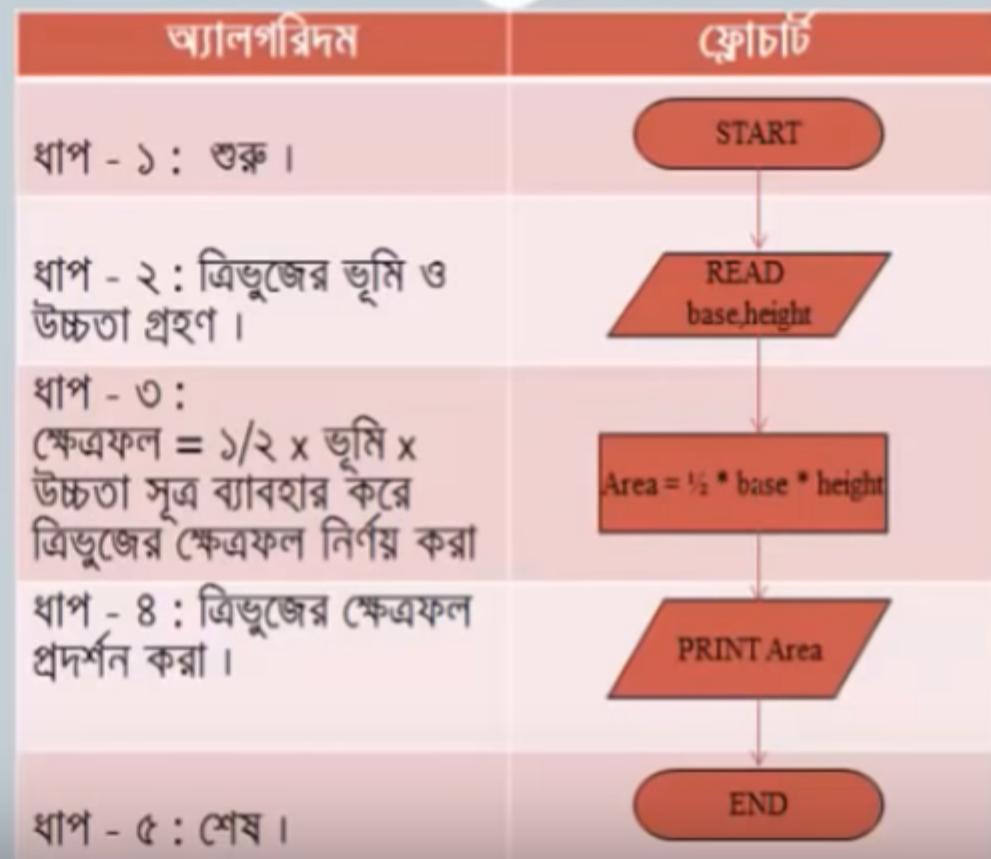
# প্রোগ্রাম তৈরির ধাপসমূহ

6

পর্ষ ৫ : একটি ত্রিভুজের ক্ষেত্রফল নির্ণয়ের অ্যালগরিদম ও ফ্রোচার্ট লিখ।

11

১. সমস্যা নির্দিষ্টকরণ
২. সমস্যা বিশ্লেষণ
৩. প্রোগ্রাম ডিজাইন
৪. প্রোগ্রাম উন্নয়ন
৫. প্রোগ্রাম বাস্তবায়ন
৬. প্রোগ্রাম রক্ষণাবেক্ষণ



# প্রোগ্রাম তৈরির ধাপসমূহ

7

১. সমস্যা নির্দিষ্টকরণ

২. সমস্যা বিশ্লেষণ

৩. প্রোগ্রাম ডিজাইন

৪. প্রোগ্রাম উন্নয়ন

৫. প্রোগ্রাম বাস্তবায়ন

৬. প্রোগ্রাম রক্ষণাবেক্ষণ

- ❖ প্রোগ্রাম কোডিং
- ❖ অ্যালগরিদিম, ফ্লোচার্টকে বা সুডোকোডকে থেকে কোন প্রোগ্রামিং ভাষায় প্রোগ্রাম লিখতে হবে।
- ❖ প্রোগ্রামিং ভাষা: C,C++,BASIC etc.

# প্রোগ্রাম তৈরির ধাপসমূহ



7

১. সমস্যা নির্দিষ্টকরণ

২. সমস্যা বিশ্লেষণ

৩. প্রোগ্রাম ডিজাইন

৪. প্রোগ্রাম উন্নয়ন

৫. প্রোগ্রাম বাস্তবায়ন

৬. প্রোগ্রাম রক্ষণাবেক্ষণ

```

2 #include <stdio.h>
3 int main()
4 {
5     float base,height;
6     float area;
7     printf("Enter base = ");
8     scanf("%f", &base);
9     printf("Enter height = ");
10    scanf("%f", &height);
11    area = 1/2 * base *height;
12    printf("Area = %.2f\n", area);
13
14 }
15

```



# প্রোগ্রাম তৈরির ধাপসমূহ

8

১. সমস্যা নির্দিষ্টকরণ



২. সমস্যা বিশ্লেষণ



৩. প্রোগ্রাম ডিজাইন



৪. প্রোগ্রাম উন্নয়ন



৫. প্রোগ্রাম বাস্তবায়ন



৬. প্রোগ্রাম রক্ষণাবেক্ষণ

- ❖ বাস্তবায়ন অংশের ২টি গুরুত্বপূর্ণ কাজ
  ১. টেস্টিং : ভুল-তরঙ্গি পরীক্ষা করা।
  ২. ডিবাগিং : ভুল সংশোধন করা।



# প্রোগ্রাম তৈরির ধাপসমূহ

9

১. সমস্যা নির্দিষ্টকরণ



২. সমস্যা বিশ্লেষণ



৩. প্রোগ্রাম ডিজাইন



৪. প্রোগ্রাম উন্নয়ন



৫. প্রোগ্রাম বাস্তবায়ন



৬. প্রোগ্রাম রক্ষণাবেক্ষণ

- ❖ দীর্ঘমেয়াদী ব্যাবহার নিশ্চিতকরণ
- ❖ হার্ডওয়্যার, সফটওয়্যার ও ডকুমেন্টশন রক্ষণাবেক্ষণ।
- ❖ কার্যকারিতা গভীরভাবে পর্যবেক্ষণ



## ডিবাগিং

10

- ডিবাগিং সম্পর্কে আলোচনা কর।
  - বাগ – প্রোগ্রামের ভুলগুলোকে বলা হয় বাগ।
  - ডিবাগিং – প্রোগ্রামে কোন ভুল পাওয়া গেলে তা সংশোধন করা হয়, যাকে ডিবাগিং বলে।
- প্রোগ্রামের ভুলগুলোকে তিনভাগে ভাগ করা হয়। যথা –
  1. সিনট্যাক্স ভুল (Syntax Error)
  2. যুক্তিসম্মত ভুল (Logical Error)
  3. তথ্য ভুল (Information Error)

# ming Bangla Tutorial 5.6 : প্রোগ্রাম তৈরির ধাপসমূহ ও ডিবাগিং

## সিনট্যাক্স ভুল (Syntax Error)

11

- সাধারণত প্রোগ্রামের ভাষায় ব্যাকরণগত ভুলগুলোকে Syntax Error বলে।

যেমন – বানান ভুল, কমা, ব্র্যাকেট না দেয়া।

```
2 #include <stdio.h>
3 int main()
4 {
5     float base, height;
6     float area
7     printf("Enter base = ");
8     scanf("%f", &base);
9     printf("Enter height = ");
10    scanf("%f", &height);
11    area = 1/2 * base *height;
12    printf("Area = %.2f\n", area);
13
14 }
15
```

# যুক্তিসম্মত ভুল (Logical Error)

12

- প্রোগ্রামে যুক্তিসম্মত যে ভুল থাকে সেগুলোকে যুক্তিসম্মত ভুল (Logical Error) বলে।

যেমন : X + Y এর পরিবর্তে X - Y লিখা ইত্যাদি।

```
2 #include <stdio.h>
3 int main()
4 {
5     float base, height;
6     float area;
7     printf("Enter base = ");
8     scanf("%f", &base);
9     printf("Enter height = ");
10    scanf("%f", &height);
11    area = 1/2 + base *height;
12    printf("Area = %.2f\n", area);
13
14 }
```

## তথ্য ভুল (Information Error)

13

- প্রোগ্রামে কোন তথ্য ভুল দিলে তাকে Information Error বা তথ্য ভুল বলে।

যেমন – কোন চলকের মান ইনপুট দেয়ার সময় ভুল মান দেয়া হয় সেক্ষেত্রে কম্পিউটার বুঝতে না পেরে ভুল ফলাফল প্রদর্শন করে।

ভূমি = ১০ মিটার

উচ্চতা = ২০ মিটার

ক্ষেত্রফল = ?

- There are also a few kinds of errors in 'C' those are not detected by the compiler.

4- Run Time Errors

5- Compile Time Errors

These error are not identified by the compiler and so they are termed as "EXCEPTIONS"

# পরিচিতি

2

## আজকের বিষয়বস্তু

১. প্রোগ্রাম কি? প্রোগ্রামের ভাষা কি ?
২. কেন এত প্রোগ্রামিং ল্যাঙ্গুয়েজ ?
৩. একটি আদর্শ প্রোগ্রামের বৈশিষ্ট্যগুলো লিখ।
৪. সি প্রোগ্রামিং কি? সি প্রোগ্রামিং এর ইতিহাস সম্পর্কে লিখ।
৫. সি প্রোগ্রামিং ভাষার বৈশিষ্ট্যগুলো লিখ।
৬. সি প্রোগ্রামিং ভাষার ব্যবহার লিখ।



## ১. প্রোগ্রাম কি? প্রোগ্রামের ভাষা কি ?

কোন সমস্যা সমাধানের জন্য কম্পিউটার ভাষায় ধারাবাহিকভাবে লিখিত কতগুলো কমান্ড বা নির্দেশের সমষ্টিকে প্রোগ্রাম বলা হয়।

প্রোগ্রাম রচনার জন্য ব্যবহৃত শব্দ, বর্ণ, অঙ্ক, সংকেত ও কতিপয় নিয়মকানুনকে একত্রে প্রোগ্রামের ভাষা বলে।

যেমন – BASIC, Pascal, C, C++, COBOL, FORTRAN, Java ইত্যাদি ।

## ২. কেন এত প্রোগ্রামিং ল্যাঙ্গুয়েজ ?

BASIC = Beginner All-Purpose Symbolic Instruction code

COBOL = Common Business Oriented Language (memo  
create,debit,credit,accounting)

FORTRAN = FORMULA TRANSLATOR (works with  
mathematical formula)

### ৩. একটি আদর্শ প্রোগ্রামের বৈশিষ্ট্যগুলো লিখ।

- প্রোগ্রাম সহজ হবে।
- প্রোগ্রাম যথাসম্ভব সংক্ষিপ্ত ও বোধগম্য হবে।
- প্রোগ্রামকে অপ্রাসঙ্গিকভাবে বড় করা যাবেনা।
- সমস্যা সমাধানের প্রক্রিয়াটি স্পষ্ট ও যুক্তি নির্ভর হতে হবে।
- কোডিং নির্ভুল হবে।
- ভবিষ্যতে সংশোধন ও পরিবর্ধনের ব্যবস্থা থাকতে হবে।
- অতিরিক্ত লুপ ব্যাবহার করা যাবেনা।

## ৮. সি প্রোগ্রামিং এর ইতিহাস সম্পর্কে লিখ।

Sr.	Language	Author	Year	Influenced by	Influenced	
1 .	ALGOL	Team of Designer	1958	Fortran	BCPL,Simula,CPL Pascal, C	
2 .	CPL	University of Cambridge and London	1960	ALGOL	BCPL	 Christopher Strachey CPI - Common Programming Language
3 .	BCPL	Martin Richard	1966	CPL	B	 Martin Richard BCPL
4 .	B	Ken Thompson	1969	BCPL	C	 Ken Thompson B Language
5 .	C	Dennis Ritchie	1972	B,ALGOL, FORTRAN	C++,C#,Java, Perl,PHP,etc.	 Dennis Ritchie

## ৫. সি প্রোগ্রামিং এর বৈশিষ্ট্য লিখ।

- I. General purpose language
- II. Middle level language
- III. System Programming Language
- IV. Structural Language
- V. Platform dependent
- VI. Case Sensitive

## ৫. সি প্রোগ্রামিং ভাষার ব্যবহার লিখ।

এই ভাষার সাহায্যে অনেক প্রোগ্রাম লিখা যায়। যেমন -

- অপারেটিং সিস্টেম
- ল্যাঙ্গুয়েজ কম্পাইলার
- ল্যাঙ্গুয়েজ ইন্টারপ্রেটার
- অ্যাসেম্বলার
- টেক্সট এডিটর
- কম্পিউটার গেমস
- কম্পিউটার ভাইরাস
- কম্পিউটার **এন্টিভাইরাস** ইত্যদি

৩.২: কি ভাষেস প্রোগ্রামং পড়বেন / পড়াবেন ?

## C programming

- ① Translator program → Compiler  
→ Interpreter  
→ Assembler
- ② Token → Keywords, Escape sequence,  
puncuators, Operators, data types
- ③ Input - output (variables  
(printf, scanf))
- ④ Operator → একটির অন্তর  
প্রত্যন্ত ৰ  
বিবরণ কৰিব  
প্রয়োজন কৰিব  
F ← C

Control statement

- Conditional (if, switch) → Even/odd  
positive/negative  
largest/smallest
- Loop control (for, while,  
do, while) → Leap year  
Small/capital  
vowel/consonant
- Flow chart

A, F, C {

1. 2. 3. 100

1 → 100 (Even)

1 → 100 (Even,

$2+4+6+8+\dots+100$

$1 \times 3 \times 9 \times \dots$

Fibonacci series

prime

gcd/lcm

Multiplication

Factorial

### List of Some Commonly used header Files and their purpose

Header Files	Purpose	Functions declared
Stdio.h	Used for standard input and Output (I/O) operations.	printf(), scanf(), getchar(), putchar(), gets(), puts(), getc(), putc(), fopen, fclose, feof()
Conio.h	Contains declaration for console I/O function	clrscr(), getch(), exit()
Ctype.h	Used for character-handling or testing character.	isupper(), islower(), isalpha()
Math.h	Declares mathematical functions and macros.	pow(), sqrt(), cos(), tan(), sin(), log()
Stdlib.h	Used for number conversions, storage allocations.	rand(), srand()
String.h	Used for manipulating strings.	strlen(), strcpy(), strcmp(), strcat(), strlwr(), strupr(), strrev()
Time.h	Used for manipulating time and date.	

Data Type	Examples	Keyword	Storage size	Format specifier
Character data	'a', 's', '1'	char	1 byte	%c
Whole numbers	125	int	4 bytes	%d
Floating point numbers	6 decimal places	float	4 bytes	%f
Double numbers	15 decimal places	double	8 bytes	%lf

- ভেরিয়েবল / ফাংশন / অ্যারে লিখার নিয়মসমূহ -
  ১. ভেরিয়েবলের নামের মধ্যে বর্ণ(A...Z, a...z), অঙ্ক(0,1,.....,9), আন্ডারস্কোর( \_ ), ডলারচিহ্ন(\$) ব্যাবহার করা যায়।
  ২. ভেরিয়েবলের নাম ডিজিট বা অঙ্ক দিয়ে শুরু হতে পারেনা।
  ৩. কোন কিওয়ার্ড, ফাংশন ভেরিয়েবলের নাম হিসাবে ব্যাবহার করা যায় না।
  ৪. ভেরিয়েবলের নামের মধ্যে কোন ফাঁকা স্থান থাকতে পারেনা।
  ৫. ভেরিয়েবলের নামকরণে সর্বাধিক ৩১ ক্যারেক্টার ব্যাবহার করা যায়। তবে ৮ ক্যারেক্টার ব্যাবহার করাই শেষ।

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

## Character Escape Sequence

## Meaning

\a	(Alert) Bell
\b	BackSpace
\f	Form Feed
\n	NewLine
\r	Carriage Return
\t	Horizontal Tab
\v	Vertical Tab
\0	Null Character
\'	Single Quote
\"	Double Quote
\\\	Backslash
\?	Questionmark

অপারেটর	কাজ	উদাহরণ	ফলাফল
+	যোগ করা	$X = 15 + 6$	$X = 21$
-	বিয়োগ করা	$X = 15 - 6$	$X = 9$
*	গুন করা	$X = 15 * 6$	$X = 90$
/	ভাগ করা	$X = 15 / 6$	$X = 2$
%	ভাগশেষ করা	$X = 15 \% 6$	$X = 3$

Assignment Operator	Example	Full meaning
=	y = x + 5 ;	
+=	x += 5 ;	x = x + 5 ;
-=	x -= y ;	x = x - y ;
*=	x *= 5 ;	x = x * 5 ;
/=	x /= 5 ;	x = x / 5 ;
%=	x %= 5 ;	x = x % 5 ;

Bitwise Operator	Meaning
&	Bitwise AND
	Bitwise OR
^	Bitwise ExOR
>>	Right shift
<<	Left shift
~	Bitwise NOT

- `abs()`
- `sqrt()`
- `pow()`
- `log()`
- `log10()`
- `exp()`
- `sin()`
- `cos()`
- `tan()`
- `round()`
- `trunc()`
- `ceil()`
- `floor()`

# Loop

- 1. For, while, do while loop syntax
- 2. Multiplication table
- 3. Factorial
- 4. Prime number
- 5. GCD,LCM
- 6. Sum of digits
- 7. Reverse number
- 8. Palindrome
- 9. Armstrong
- 10. Counting number of digits in an integer
- 11. Strong number

- $1 + 2 + 3 + \dots + n$
- $2 + 4 + 6 + 8 + \dots + n$
- $1.5 + 2.5 + 3.5 + \dots + n$
- $1^2 + 2^2 + 3^2 + \dots + n^2$
- $1 \times 2 \times 3 \times \dots \times n$
- $1^2 \times 2^2 \times 3^2 \times \dots \times n^2$
- $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$
- Fibonacci series
- Lucas series

**Pattern-1**

N=3

1  
1 2  
1 2 3

**Pattern-3**

N=3

1  
1 0  
1 0 1

**Pattern-5**

N=3

A  
A B  
A B C

**Pattern-7**

N=3

\*

\*\*

\*\*\*

**Pattern-2**

N=3

1  
2 2  
3 3 3

**Pattern-4**

N=3

1  
0 0  
1 1 1

**Pattern-6**

N=3

A  
B B  
C C C

**Pattern-8**

N=3

#  
##  
###

Number right angle triangle

Binary right angle triangle

Alphabetic right angle triangle

Pattern-1

N=3

1 2 3

1 2

1

Pattern-3

N=3

1 0 1

1 0

1

Pattern-5

N=3

A B C

A B

A

Pattern-7

N=3

\* \* \*

\* \*

\*

Pattern-2

N=3

3 3 3

2 2

1

Pattern-4

N=3

1 1 1

0 0

1

Pattern-6

N=3

C C C

B B

A

Pattern-8

N=3

# # #

# #

#

# Pattern Type-4

Pattern-1

N=3

1  
1 1  
1 2 3

Pattern-3

N=3

1  
1 0  
1 0 1

Pattern-5

N=3

A  
A B  
A B C

Pattern-7

N=3

\*

\*\*

\*\*\*

Pattern-2

N=3

1  
2 2  
3 3 3

Pattern-4

N=3

1  
0 0  
1 1 1

Pattern-6

N=3

A  
B B  
C C C

Pattern-8

N=3

#  
##  
###

# Pattern Type-5

Pattern-1

N=3

1 2 3

1 2

1

Pattern-3

N=3

1 0 1

1 0

1

Pattern-5

N=3

A B C

AB

A

Pattern-7

N=3

\* \* \*

\* \*

\*

Pattern-2

N=3

3 3 3

2 2

1

Pattern-4

N=3

1 1 1

0 0

1

Pattern-6

N=3

C C C

B B

A

Pattern-8

N=3

# # #

# #

#

# Pattern Type-6

14

Pattern-1
N=3
1
1 2
1 2 3

Pattern-2
N=3
1 2 3
1 2 3
1 2 3
1 2
1

Pattern
N=3
1
1 2
1 2 3
1 2 3
1 2
1

Pattern-1 N=3	Pattern-2 N=3	Pattern-3 N=3	Pattern-4 N=3	Pattern-5 N=3
1	1	A	A	*
1 2	2 2	AB	B B	**
1 2 3	3 3 3	ABC	CCC	***
1 2	2 2	AB	B B	**
1	1	A	A	*

# Pattern Type-7

Pattern-1

N=5

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

Pattern-2

N=5

1 1 1 1 1  
2 2 2 2 2  
3 3 3 3 3  
4 4 4 4 4  
5 5 5 5 5

Pattern-3

N=5

1 2 3 4 5  
1 2 3 4 5  
1 2 3 4 5  
1 2 3 4 5  
1 2 3 4 5

Pattern-4

N=5

A A A A A  
B B B B B  
C C C C C  
D D D D D  
E E E E E

Pattern-5

N=5

A B C D E  
A B C D E  
A B C D E  
A B C D E  
A B C D E

# Pattern Type-8 (Pyramid-I)

Pattern-1

N=3

\*  
\*\*\*  
\*\*\*\*\*

Pattern-2

N=3

1  
2 2 2  
3 3 3 3 3

Pattern-3

N=3

1  
1 2 3  
1 2 3 4 5

Pattern-4

N=3

A  
B B B  
C C C C

Pattern-5

N=3

A  
A B C  
A B C D E

# Pattern Type-9

Pattern-1

N=3

\*\*\*\*\*

\*\*\*

\*

Pattern-2

N=3

3 3 3 3 3

2 2 2

1

Pattern-3

N=3

1 2 3 4 5

1 2 3

1

Pattern-4

N=3

C C C C C

B B B

A

Pattern-5

N=3

A B C D E

A B C

A

# Pattern Type-10

22

<p>Pattern-3 N=3</p> <pre> * *** *****</pre>	<p>Pattern-1 N=3</p> <pre>* *** ***** ***** *** *</pre>	<p>Pattern-1 N=3</p> <pre>* *** ***** ***** *** *</pre>	<p>Pattern-2 N=3</p> <pre>1 222 33333 222 1</pre>	<p>Pattern-3 N=3</p> <pre>1 123 12345 123 1</pre>	<p>Pattern-4 N=3</p> <pre>A ABC ABCDE ABC A</pre>
<p>Pattern-4 N=3</p> <pre>***** *** *</pre>					

# 1D Array

1. What is Array?
2. Types of Array
3. Declaration, Initialization , sum of Array
4. Sum and Average of an Array
5. Maximum and Minimum of Array
6. Fibonacci series using array
7. Searching a number (Linear search)
8. copy all elements of an array to another array

# 2D Array

1. Introduction to 2D Array
2. Getting input for 2D Array
3. Simple Matrix
4. Matrix Addition & Subtraction
5. Matrix Multiplication
6. Matrix Transpose
7. Sum of diagonal elements of a matrix
8. Sum of upper & lower triangles elements

# String functions

- `strlen(str)` – To find length of string str
- `strrev(str)` – Reverses the string str as rts
- `strcat(str1,str2)` – Appends str2 to str1 and returns str1
- `strcpy(st1,st2)` – copies the content of st2 to st1
- `strcmp(s1,s2)` – Compares the two string s1 and s2
- `strcmpi(s1,s2)` – Case insensitive comparison of strings

# End of string Topic

- Introduction to String
- Input and display String
- Display string character-wise
- String swapping
- Palindrome string
- Number of vowels, consonants, words, digits and others
- Number of capital-small letters, digits.
- strlen()
- strcpy()
- strrev()
- strupr()
- strlwr()
- strcmp()
- strcat()



## Part-11 : Function

---

1. Sum of 2/3 numbers
2. Square of a number
3. Area of a triangle
4. X to the power Y
5. Passing array to a function
6. Finding maximum / minimum value from an array using function
7. Recursion
8. Factorial using recursion

# Custom data type

1. structure
2. union
3. enum
4. typedef