

- ① Introduction to programming
- ② Input -Output.
- ③ Operators
- ④ Math
- ⑤ control statement → if, else
- ⑥ switch
- ⑦ Loop
- ⑧ series
- ⑨ Pattern/ Pyramide
- ⑩ Array (one dimensional)
- ⑪ Array (two dimensional)
- ⑫ String
- ⑬ Function
- ⑭ Recursion
- ⑮ File
- ⑯ Structure
- ⑰ Pointers
- ⑱ Advance C

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

```
int main()
```

```
{
```

```
char ch;
```

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

```
switch (ch)
```

```
{
```

```
case ('a'):
```

```
case ('e'):
```

```
case ('i'):
```

```
case ('o'):
```

```
case ('u'):
```

```
    printf("vowel");
```

```
    break;
```

```
case ('a'):
```

```
    printf("vowel");
```

```
    break;
```

```
case ('e'):
```

```
    printf("vowel");
```

```
    break;
```

```
default
```

```
    printf("consonent");
```

```
}
```

# Topic - 1 → introducing to programming.

c-1 to c-7

## ① Translator Program

- ① compiler
- ② Interpreters
- ③ Assemblers

## ② Token →

keywords, Escape sequence, Punctuation  
operators, data types, variables.

## ③ Input, Output

(printf, scanf)

## ④ Operator

অঙ্গুলি কোড়াল, { কোড়াল Type }  
সময়ের " "  
সামান্যতের " "

কান্দার মান ও কাত

convert type (c → f, f → e, centim  
m → km)

## ⑤ condition control statement

i) conditional (if, else, switch)

ii) loop (for, while, do, while)

Even/Odd  
positive/Negative  
large

principles of programming

F-9 GK I-9

## ⑤ control statements

① conditional (if, else, switch)

② loop (for, while,  
do while)

A, F

loop flowchart

flowchart

even / odd  
positive / negative  
largest / smallest  
Leap year  
small / capital  
vowel / consonant

## ⑥ Array

## \*⑦ Function

target, target

target, target

target

target

target

target

target

target

target

Program: যেন্তা প্রযোজনীয় কর্মকে নির্দিষ্ট করে।

লেখাপ্ত ০.১ লিখিত নির্দেশের সম্মিলনে প্রোগ্রাম লে।

Programming language: প্রোগ্রাম করার জন্য বিশ্বিত

বর্ণ, অঙ্ক, অস্থোত্তর এবং এন্টেন্ডেড বিক্ষিয়ে নিয়ম মিলিয়ে  
তৈরি করা হ্যাত প্রযোজ্ঞিত ভাষা।

Ex. C, C++, Assembly language, Java

Translator Program: আমাদের ভাষা/ Programming ভাষাকে

computer এর ভাষায় নিয়ে যাওয়ার জন্য একটা একটা একটা হ্যাত  
translator program/ অনুবাদক প্রযোজ্ঞ।

\* উচ্চতর ভাষায় লিখিত স্কেচ কে প্রযোজ্ঞ করে source  
code করে।

\* source code কে computer এর ভাষা (বাইনারি)-হ্যাত

নিয়ে যাওয়ার জন্য ভাইন মেশিন কোডে করে।

বা object code

\* অনুবাদক প্রযোজ্ঞ করে  $\rightarrow$  source code কে object  
code করে নিয়ে যাওয়া,

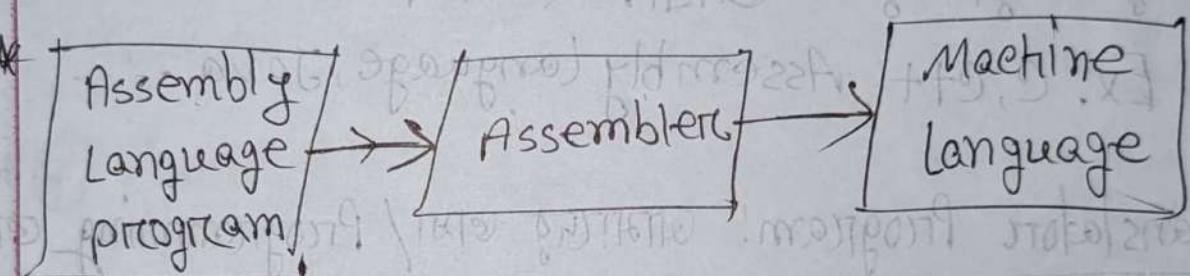
ব) অনুবাদক প্রোগ্রাম-৩ প্রকার :

i) Assembler (অ্যাসেম্বলার)

ii) compiler (কমাইলাৰ)

iii) interpreter (ইন্টেপ্ৰিউটাৰ)

ব) Assembler:

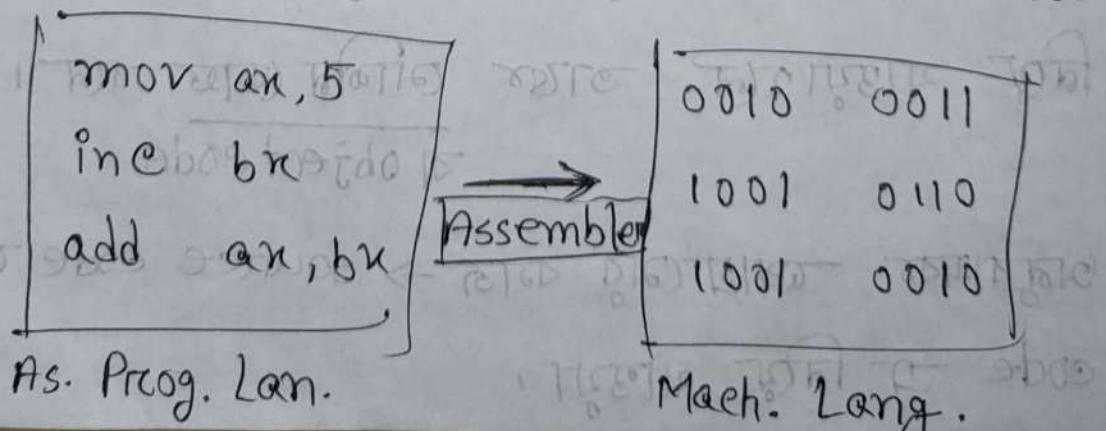


\* Assembly Programming language কে Machine

Language এ রূপান্তৰ কৰে Assembler.

\* এ অনুবাদক Assembly আধাৰ লিখিত এ প্ৰোগ্ৰামৰে

মেশিন আধাৰ রূপান্তৰ কৰতে তাৰে Assembler বল।



\* move → mov { shortcut কর্তৃপক্ষ লেখা হ্যাঁ।  
 addition → add { এবং এদেশ কলা হ্যাঁ নমোনিক।  
 increment → inc } mnemonic.

### \* মুর্বি, অঙ্গুরি:

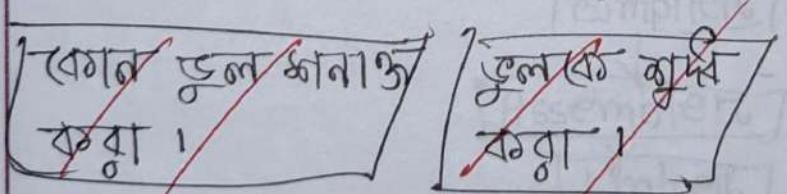
① নমোনিক কোডের মীলিন জাখাপু বুপান্ত করে।

② Program এ একান তুল মাকলে Execute

মোড়ের show কর্তৃপক্ষ রেসেভ সেস্টেম:

③ Program এর তুল মাইক্রোসিন কর্তৃপক্ষ।

④ ডিবাগিং ও টেস্টিং এর কাজ দ্রুত গতি সম্পন্ন।



⑤ Memory টে বসা থাপুণা হ্যাঁ।

\* অ্যাবৰ্ধিং

i) Program নির্যাপ্ত করা অবেক্ষণ কৈশী মাধ্যম  
লাগে।

computer এর তুলনামূলক এয়া 5-25 মন মাধ্যম কৈশী  
কৈশী লাগে।

\* প্রথমে Program আসাকে compile করা হয়—  
use

C, C++, objective-C, C#, Pascal, COBOL, ADA,  
visual Basic, Smalltalk, Scheme

\* প্রথমে program দেখাকে interpret করা হয়—

Basic, PHP, Python, perl, Ruby, Java-

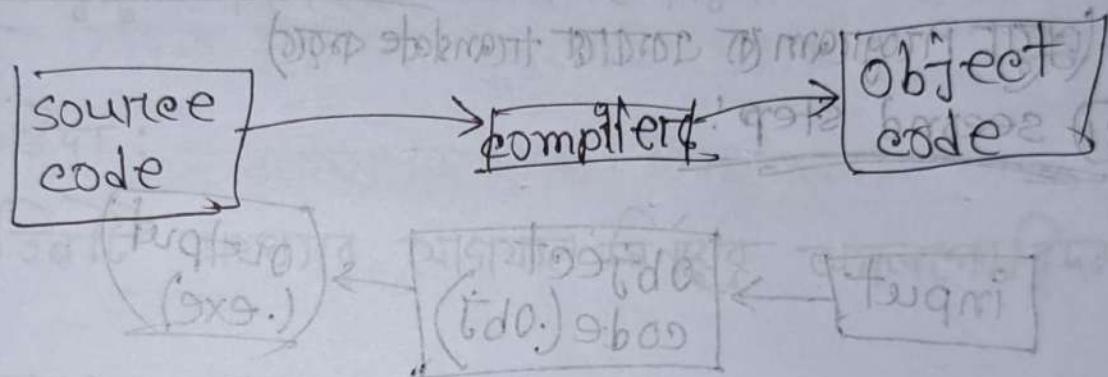
Javascript

অনুবাদক

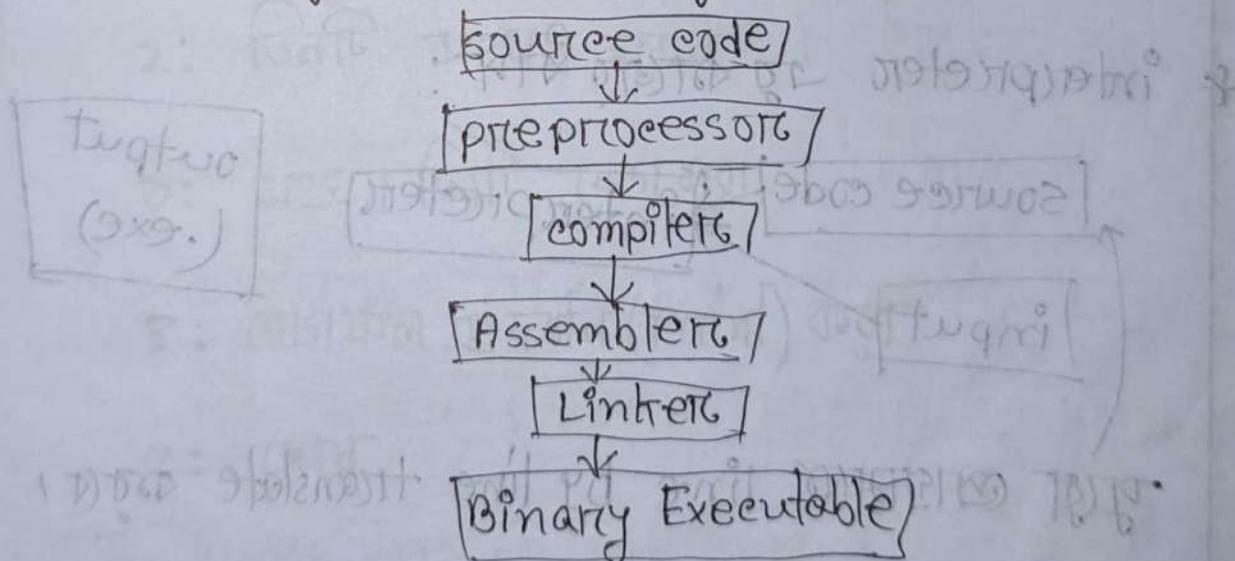
## \* compilers:

এর অনুবাদক প্রোগ্রাম source code এর object code হিসাবে কাট্টি তাছে compiler

যান।

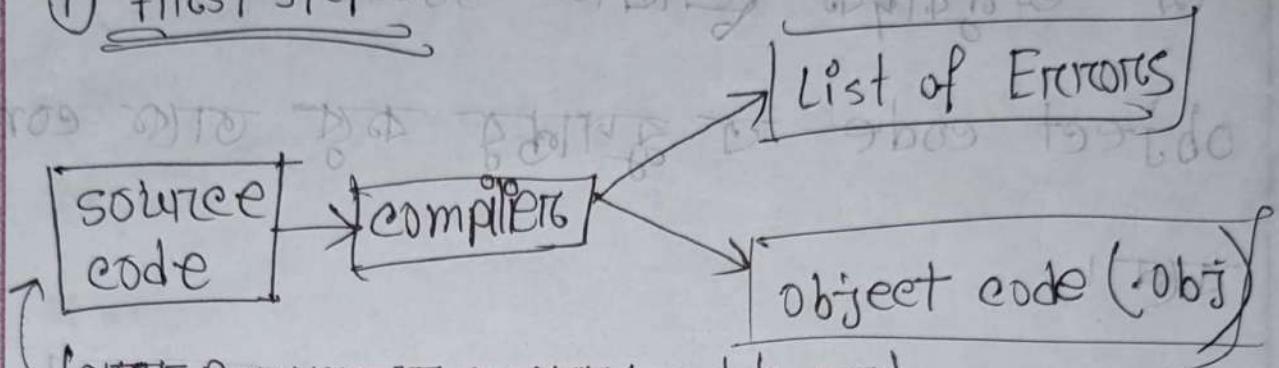


## \* Language processing System:

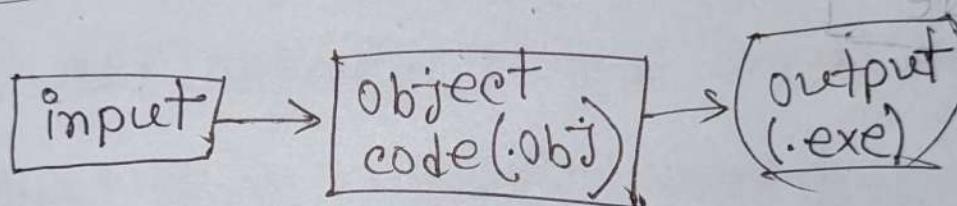


\* compiler পদ্ধতি (২ হিসেবে কার্য)

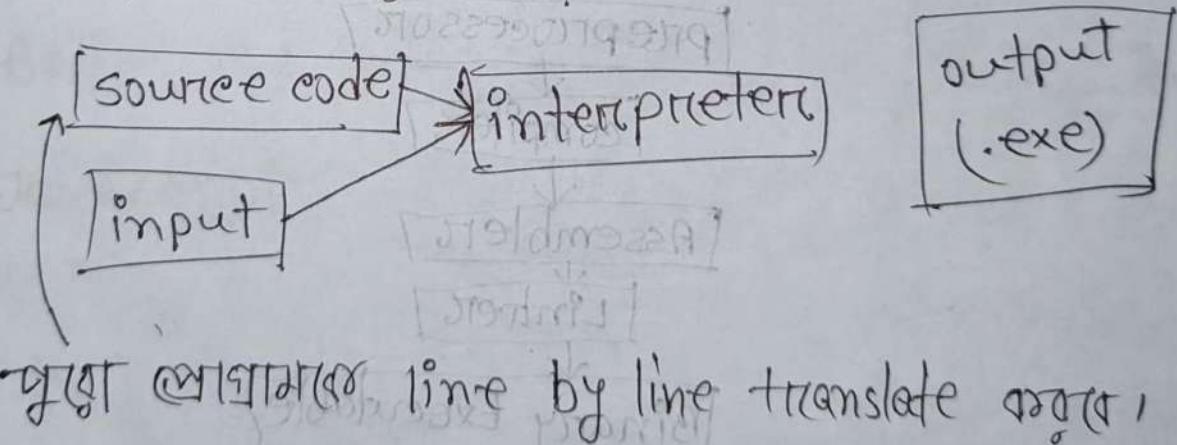
i) first step



ii) second step



\* interpreter এর কাজের ধীপ:



পুঁজি প্রোগ্রামকে line by line translate করে।

C-3

Algorithm

Algorithm

আলগোরিদম

Algorithm:

ক্ষেত্র মাত্রাকে বিশেষভাবে প্রয়োজন কৃত্যে আলগোরিদম  
বর্ণাবৃত্তি নথিতে আলগোরিদম বলে।

১০১ হাতুর কাসীটে শুভ্রাংশু পোড়া পোড়া

\*উদাঃ

⇒ তিনটি মন্দ্যাবৃত্তি মাত্রাকে বিশেষভাবে আলগোরিদম লেখ-

ক্ষেত্র-১: ক্ষেত্র

২: তিনটি মন্দ্যা ত্বরণ

৩: মন্দ্যা তিনটি যোগ করা

৪: মাত্রাকে অদর্শন (Print) করা

৫: শেষ

ক্ষেত্র প্রস্তুত করিবি। ক্ষেত্র প্রস্তুত করিবি।

## \* নিয়ম ও টেক্নিক্সঃ

- ① গোলগোলিম মাইক্রোফোন রেজ হবে।
- ② অত্যক্ষ ধীপ অর্থ রেজ হবে। এল মাত্রে  
গোল Program অর্থেই সুন্দর পাও।
- ③ ধীপে বি ধীপে অভ্যন্তর ঘৰ্মধীন কষ্টের হবে।
- ④ এসে কানের ওপরে প্রয়োগ কর্তৃত উৎসোগী কষ্টের  
হবে।

## \* সুবিধা:

- ① মাইক্রো Program এর উদ্দেশ্য সুন্দর আশুর  
কর্তৃ।
- ② Program এর কোন নির্ভয় পাশ্চায় কর্তৃ।
- ③ Prog. পরিবর্তন, পরিবর্তনে সামান্য কর্তৃ।
- ④ এসের গোল Program. লিখতে সামান্য কর্তৃ।

\* পিতৃজ্যু ক্ষেত্রফল নির্ণয় এ আলগোড়িম-

ধীপ-১: শুনু ।

২: পিতৃজ্যু ক্ষেত্রফল ও উচ্চতা গ্রহণ ।

৩:  $\text{ক্ষেত্রফল} = \frac{1}{2} \times \text{ভূমি} \times \text{উচ্চতা ঘূর্ণাবলী}$

কর্তৃ পিতৃজ্যু ক্ষেত্রফল নির্ণয় করা ।

৪: ক্ষেত্রফল অদর্শন করা ।

৫: শেষ ।

\* ২টি পূর্ণ মাত্রায় মার্খ বচ্ছে মাত্রায় কর্তৃ কর্তৃ আলগোড়িম-

ধীপ-২: শুনু ।

২: অত্যন্ত ২টি গ্রহণ ।

৩: এম-অত্যন্তাটিক্রমে কি ২য় মাত্রায় হতে বচ্ছে ?

(ক) হ্যাঁ, ফলাফল ছাপি এম-অত্যন্তাটি বচ্ছে ।

(খ) না, " " " " "

৪: শেষ ।

## Flowchart

### ফ্লোচ-চার্ট:

Program এর যদি চিত্র/প্রতিক্রিয়া মাধ্যমে এবং  
কর্তৃপক্ষ দ্বারা তাn Flowchart পর্যন্তের Flowchart  
বলে।

Ex.

\* তিনটি মত্থাগুরু যোগফল বিরচ্ছে এড় Algorithm & Flowchart  
লিখ। →

Algorithm

ধীপ-১: ক্ষেত্র।

২: তিনটি মত্থাগুরু করা।

৩: মত্থা তিনটি যোগ করা।

৪: যোগফল প্রদর্শন করা।

৫: মোহর।

Flowchart

(Start)

READ A,B,C

SUM A+B+C

PRINT SUM

(END)

\* Flowchart କିମ୍ବା ?

① system

② program

\* System Flowchart system କିମ୍ବା ?

\* System Flowchart: ଯେ Flowchart ଏହା ଏକାଟି system

ଏହା ତେଣୁ ଆଧାର ଅନ୍ତିମାବ୍ୟନ, ମଧ୍ୟକାଳୀନ ଓ ପରିପାଲନ

ଅଧିକାରୀଙ୍କ ଦିକ୍ଷା ଚିନ୍ତିତିରେ ତାଙ୍କେ system Flowchart

ଦିଲେ ।

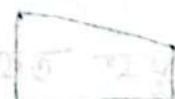
ବା Programming ଏହାଙ୍କା

\* Program Flowchart: Program ପ୍ରକାରମୁକ୍ତ ଏହା

Flowchart କୌଣସି ରୂପାବଳୀ ତାଙ୍କେ Program Flowchart

ଦିଲେ ।

ମିନିକ ପ୍ରୋଗ୍ରାମ୍ ହିନ୍ଦୁକିମ୍ବା



ଆଧାରମୁକ୍ତ ଏହାଙ୍କା ହେବାରେ କିମ୍ବା କିମ୍ବା



ଫୁଲ୍ କିମ୍ବା କିମ୍ବା



ଆଧାରମୁକ୍ତ ଏହାଙ୍କା ହେବାରେ କିମ୍ବା କିମ୍ବା

ଏହା ପରିପାଲନ କାମକାଳୀନ ଏହାଙ୍କା ହେବାରେ

## \* System Flowchart ରେ ଅଣିରୁ:

[ ] → ଅନ୍ତିମରୂପ

[ ] → ଡାଟା ମେଲ୍

( ) → Magnetic tape

[ ] → disk

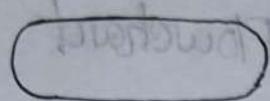
[ ] → card

[ ] → online ମୂଳି

[ ] → Manual input

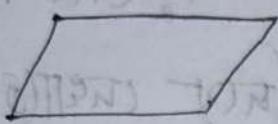
◇ → ଆଟି ବା ଧାଜାନୀ

## \* Programming Flowchart:



→ টার্মিনাল প্রতীক, তিথেক আছে। (START/END)

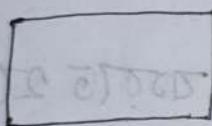
Prog. এতু শব্দ, শেখ করতে USE হয়।



→ গুণ, নির্ভাস প্রতীক, মাত্রিক আছে।

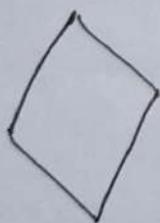
Prog. এতু input, output করতে USE হয়।

(INPUT/OUTPUT)



→ অক্ষিয়াকরণপ্রতীক, আয়তআকৃতি। Prog. এতু

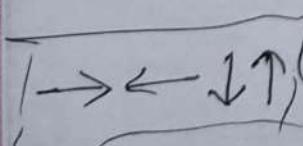
অক্ষিয়াকরণে জন্ম ব্যবহৃত হয়। (গানিকা  
চিয়ার-নিয়মাবলী এবং বিভিন্ন বিকল্প বিকল্প অক্ষিয়াকরণ)



→ নির্ণয় প্রতীক। রীত্বক/diamond আছে।

নির্ণয় প্রয়োগে জন্ম ব্যবহৃত হয়। এতু ২টি

সম্ভাবনা থাই। হ্যাঁ ও না।



Prog. এতু দিচ ব্যাখ্য।



এস্যার প্রতীক। এক উপর্যুক্ত বক্তৃ Flowchart  
এতু গাফদা না যান প্রতি কাশার করতে বাকি  
অঙ্ক পাবে ম্যাচ সঁওয়া হয়।

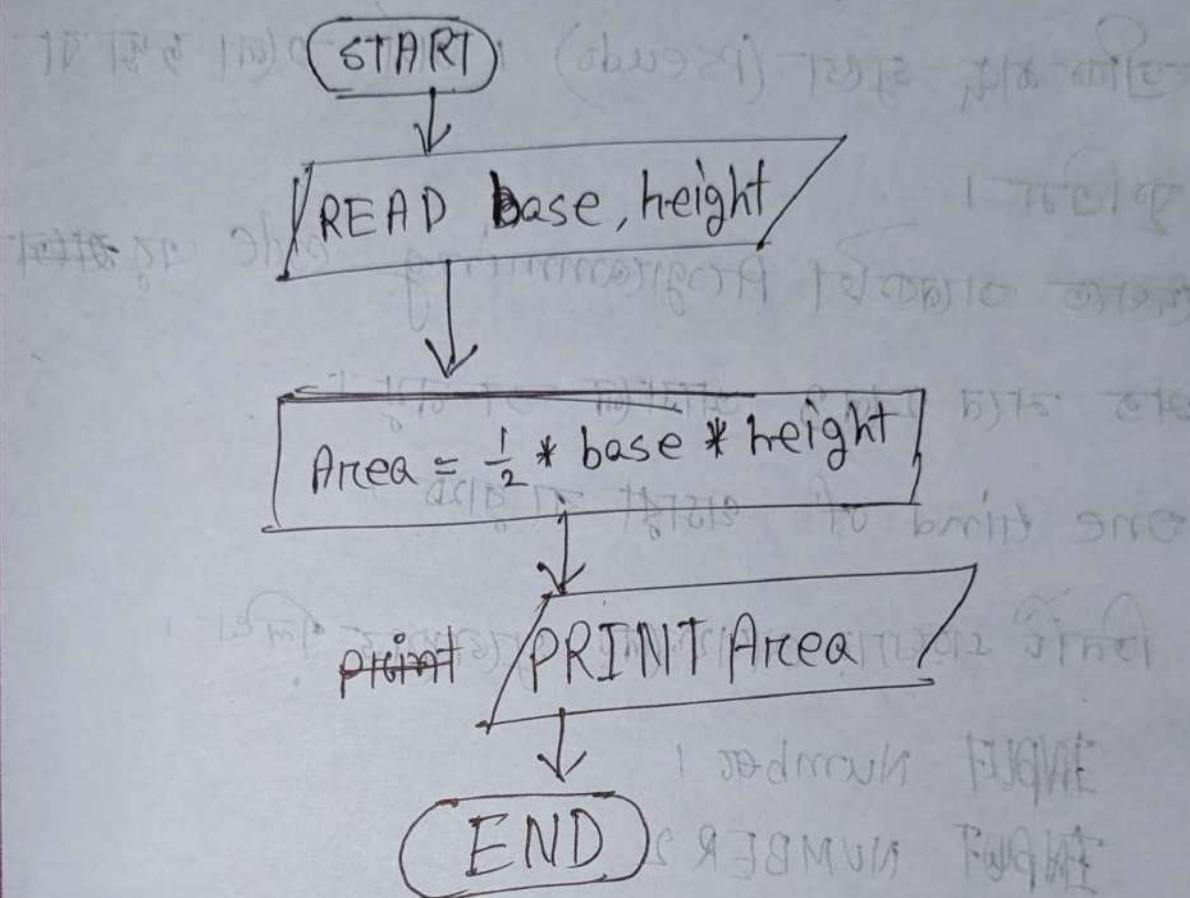
## \* Flowchart ଜୀବନାର୍ଥ ନିର୍ମାଣ:

- ① ବିଦେଶ କିଛି ଏତୀକା କାବ୍ୟାରୁ କହୁଁ Flowchart  
ଆଗତେ ହେ ।
- ② Flowchart କ୍ଷେତ୍ରରେ ଦିଲ୍ଲି - ହୃଦୟ ଉପରେ ନିଚ୍ଚ  
ଅଧିକ ସାମ ହାତେ ଆଜି ତୀର୍ତ୍ତ କିମ୍ବ ଦିଲ୍ଲି ଦେଖିବାରେ ।
- ③ ଏତିକ୍ଷେତ୍ରେ ହେଉଁ ଲେଧାନୁଲୋ ମାଝେ ଏହା ଅହଳକାରୀ  
ହାତ ରହି ।

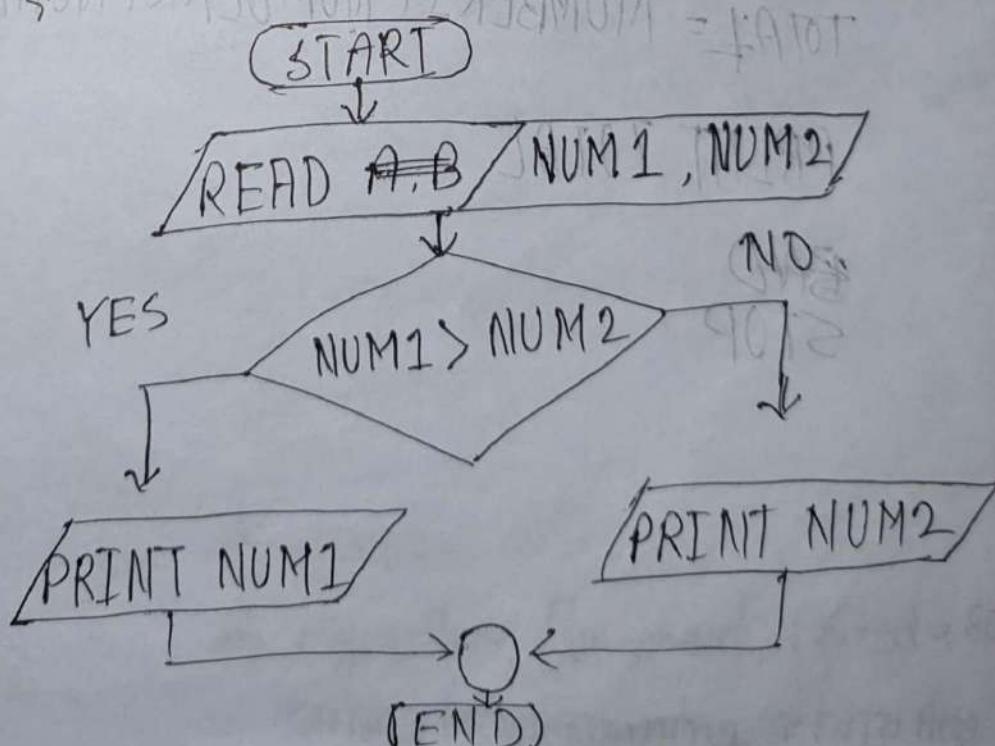
- ④ ଅନିବିକ୍ରି ଏତୀକ୍ଷେତ୍ରେ ବାବ୍ୟାରୁ ପରିଚାର ବାହାର ରହିବାରେ ।
- ⑤ ବିଦେଶ ଏବଂ ଧ୍ୟାନ ପ୍ରାପ୍ତାବିତ୍ତ ଲୋକା ବାବ୍ୟାରୁ କହା ଜୀବିତ  
ବନ୍ଧୁ ।



\* ମିଟ୍‌ଜାର୍ କୁଣ୍ଡଳ ବିନ୍‌ଧୂ ଫ୍ଲୋଚାର୍ଟ.



\*  $2\pi$  ପରିପ୍ରକାର ଦି ଶେଯି ହେବାରେ ଫ୍ଲୋଚାର୍ଟ.



ଏହି ପ୍ରୋଗ୍ରାମ୍ (Pseudo code)

\* କ୍ରମିକ ଶବ୍ଦ ପ୍ରୋଗ୍ରାମ୍ (Pseudo) । ଆମା ଲାଇ | ଛାପ ରା

କ୍ରମିମା ।

\* ଦେଖାତେ ଅନେକଟି ପ୍ରୋଗ୍ରାମିଙ୍କୁ କ୍ରମିମା ଏହାହାତି

ଏହି ମାତ୍ର ସାରେ ଯାଇବା ଆମାଙ୍କ ତା ନାହିଁ ।

\* One kind of ଧରଣ ରାଶିରେ ।

\* କିମ୍ବା ଏହାହାତି ଯେବେଳେ ପ୍ରୋଗ୍ରାମ୍ ଲିଖି ।

INPUT Number 1

INPUT NUMBER 2

INPUT NUMBER 3

TOTAL = NUMBER1 + NUMBER2 + NUMBER3

PRINT TOTAL

END

STOP

SMUIN LIKIP

SMUIN LIKIP

(END)

c-5

## কু প্রোগ্ৰাম তৈরীতি বিষয়ানুশ ও ক্ষেত্ৰিক:

Program এতু আবাধে কোন যোগ্য অমূল্যীন কষ্ট নহ'।  
যোগ্য অমূল্যীন ক্ষেত্ৰে কৃত্য কৃত্যুলোভে বিশ্ব অনুসৃত কৃত্যো হ'।  
প্ৰোগ্ৰাম বিশ্ব কৃত্যুলোভে প্ৰোগ্ৰাম তৈরীতি বিশ্ব কৃত্যো।

### বিশ্ব অনুশ-

#### ① যোগ্য নিৰ্দিষ্টকৃতি-

- (i) এখন বিশ্ব অনুশ
- (ii) " যোগ্য নিৰ্দিষ্টকৃতি
- (iii) Prog. design
- (iv) " যোগ্য তথ্য গঠণ

#### ② যোগ্য বিশ্ব-

- (i) input, output - ক্ষেত্ৰ অনুসৃত
- (ii) computer এতু যোগ্য নিৰ্বাচন কৰা

#### ③ প্ৰোগ্ৰাম উজাই-

- (i) অ্যালগোরিদম
- (ii) ফ্ৰাঞ্চেচ

#### ④ প্ৰোগ্ৰাম ভিত্তি-

- (i) প্ৰোগ্ৰাম কোড়ি
- (ii) Algorithm, Flowchart, Pseudocode থেকে  
কোন Programming কোড়ু যোগ্য Prog. লিখত হ'স।

### ⑤ প্রশ্নাম বাস্তুবাধন:

- ① বাস্তুবাধন - অংশে ২ টি শুল্কসমন্বয় করা
- ② চৈমিতি - ফুলকুটি পরিষ্কা করা
- ③ তিবিগিঙ্গ - " অংশের ছৃঞ্জ

### ⑥ প্রশ্নাম বৃক্ষনাশকরণ:

\* দীর্ঘমেয়াদী এবং নিষ্ঠ।

\* ২৫৩৭৮ মূল্যে ৩ ডাক মুল  
বৃক্ষনাশকরণ

\* এককারিতা গতীয়তে পর্যবেক্ষণ

বৃক্ষচোক - tuftus, tuomi.

বৃক্ষচোক শুধু মুক্তিক্ষেত্র

বৃক্ষচোক গোপনীয় ।

গোপনীয়

বৃক্ষচোক গোপনীয়

বৃক্ষচোক গোপনীয়

বৃক্ষচোক গোপনীয়

## କ୍ଷିତିବିଗଣିତ

\* ସାହୁ - ପ୍ରୋଗ୍ରାମ୍ରୂପ ତୁଳନାଲୋକେ ବାଗରେ ଥିଲା ।

\* ତିବିଗଣିତ - ପ୍ରୋଗ୍ରାମ୍ କୋଣ ତୁଳନାଲୋକେ ଆଧିକାରିଙ୍କ କମ୍ପ୍ୟୁଟର, ଯାଏଇ ତିବିଗଣିତ ବଳେ ।

4 ପ୍ରୋଗ୍ରାମ୍ ଓ ସିଫ୍ଟ୍‌ସ୍ଵରୂପରେ ତୁଳନାରେ ଥାଏ -

- ① ସିନଟକ୍ୟୁଟ୍‌ରୋତ୍ତ୍ରୀ (Syntax Error)
- ② ଲ୍ୟୁଜିମାର୍କ୍‌ତୁଳନା (Logical Error)
- ③ ତଥାତୁଳନା (Information Error)

① \* Programming ଏହି ଶବ୍ଦରୁପ - ସିଫ୍ଟ୍‌ସ୍ଵରୂପରେ ତୁଳନାଲୋକେ Syntax Error ବଳେ ।

Ex. ସାରାତୁଳନ, କମାତ୍ରାଖେ ନାହୁଥିଲା ।

② \* Program ଏ - ଲ୍ୟୁଜିମାର୍କ୍‌ତୁଳନା ମାତ୍ରରେ ନାହିଁ  
ଯାଏଇ ଗେଡ଼ି ଲ୍ୟୁଜିମାର୍କ୍‌ତୁଳନା ।

Ex.  $x + 2$  ଏହି ଫାଂକ୍ଷନାକୁ  $x - 2$  ଲେଖିଥିଲା ।

### ③ Program এ কোন তথ্য তুলে দিলে আগে Information Errors রয়ে।

Ex. Input তুলে দিলে মানুষৰ তুল যাবত।

\* There is also a few kinds of errors in  
'e' those is not detected by the  
computer.

④ Run time errors

⑤ compiler time errors

these is not identified by the compiler and

so they is termed as "Exception".

বেশি সহজে উলিপুরুষ হওয়া হচ্ছে

প্রকৃতি উলিপুরুষ হওয়া হচ্ছে

বেশি T-X প্রাপ্তি হওয়া হচ্ছে

C-G

ଉଦ୍‌ଦେଶ୍ୟ ଅମାଦା ଯାମଧିନେତ୍ରେ ଗଲା ବରମିତିଟିଟ୍ରେସ୍ କଷାୟ  
ଧୀର୍ଘବିକଳାରେ ଲିଖିତ କରନ୍ତୁଲୋ କମାନ୍ତ ବା ନିର୍ଦ୍ଦେଖେଁଟ୍ରେସ୍  
ମାର୍କିଟିଙ୍ ପ୍ରାଗ୍ରାମ ବଳ ।

\* ପ୍ରାଗ୍ରାମ ଏନ୍ଥେ ଗଲା କରନ୍ତୁ କାବ୍, ବର୍ତ୍ତ, ଆଶ୍ରିତ ଯାତ୍ରକ୍ରିୟା ଓ ବାଚିମୁଦ୍ରା ନିଯୁମଳନ୍ତୁକ୍ରିୟାରେ ପ୍ରାଗ୍ରାମେ ଲେଖା ବଳ ।

Ex. BASIC, Pascal, C, C++, COBOL, FORTRAN, JAVA...

ଉଦ୍‌ଦେଶ୍ୟ ଅମାଦା ଯାମଧିନେତ୍ରେ ଗଲା ବରମିତିଟିଟ୍ରେସ୍ କଷାୟ  
BASIC - Beginner All-Purpose Symbolic Instruction  
code.

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

FORTRAN - Formula Translators (works with  
mathematical formulae)

ଏହା ଏକାଟି ଆଦର୍ଶ ପ୍ରାଚ୍ୟାମନ୍ତ୍ର ବିଳିକି:

\* ପ୍ରାଚ୍ୟାମନ୍ତ୍ର ଲାଗୁ ୨୮

\* " ଅଧ୍ୟାତ୍ମାତର ଅଧିକାରୀ ଓ ପ୍ରେରିଜାମା ୨୮ ।

\* " ଅପ୍ରାଚ୍ୟାମନ୍ତ୍ର ବୃଦ୍ଧ ବୃଦ୍ଧ ଥାଏ ନା ।

\* ଆମା ଧର୍ମାନ୍ତ୍ରରେ ଏହୁମାଟି ଅଧିକ ଓ ଯୁଦ୍ଧିଲିଙ୍ଗେ  
ଲାଗୁ ୨୮ ।

\* କ୍ରାତିତ୍ଵ ନିଷ୍ଠା ୨୮ ।

\* ତ୍ରୈଷ୍ୟାତେ ଧର୍ମାନ୍ତ୍ରରେ ଓ ମହିଂଶୁ କରିଲୁ ବାଧା  
ଲାଗୁ ।

\* ଅତିବିଜ୍ଞ ଲୁପ୍ତ ହେବନ୍ତା ବୃଦ୍ଧ ଥାଏ ନା ।

(ମୁଖ୍ୟମାନ୍ତ୍ର ଲୁପ୍ତ ହେବନ୍ତା ବୃଦ୍ଧ ଥାଏ ନା ।)

(ମୁଖ୍ୟମାନ୍ତ୍ର ଲୁପ୍ତ ହେବନ୍ତା ବୃଦ୍ଧ ଥାଏ ନା ।)

(ମୁଖ୍ୟମାନ୍ତ୍ର ଲୁପ୍ତ ହେବନ୍ତା ବୃଦ୍ଧ ଥାଏ ନା ।)

## ଉ ୧ C-ଶ୍ରାଣତାବିହିନୀ ପରେ କେବଳିରେ-

\* General purpose language.

\* Middle level language

\* System programming language.

\* Structure lang.

\* Platform dependent

\* Case sensitive

## ଉ ୨ C-Programming ପରେ କଥାରେ-

① ଏହୁ ମାଧ୍ୟମରେ ଆନନ୍ଦ ପ୍ରୋଗ୍ରାମ ଲିଖିବା ଯାଇଁ । ଯାମରେ

\* ଗୋଟିଏ ଫିଲେଟର

\* Language compiler

\* "    Interpreters.

\* Assemblers.

\* Text Editors

\* Computer Games

\* virus Antiviruses.

F.9

C-7

মন কোডিং এবং যা যান্তরে,

\* এডিটর  
\* compiler  
\* ডিভাগার্স } IDE - Integrated Development Environment.

মন বিশ্লেষণে IDE -

- \* Eclipse
- \* Code::Blocks (open source, cross platform)
- \* NetBeans
- \* code lite
- \* Qt creator
- \* Dev c++
- \* #include <studio.h>  
int main

c-8

(5.10)

## topic - 2 → input - output

c-8 do c-

\*#include <stdio.h>

1 # include <stdio.h>

2 int main()

3 {

4 printf(" My name is Anisul Islam Rubel");

5 return 0;

6 }

7 }

\*@ Program এং কেজুন্ট Print f() নাম একটি লাইব্রেরী

ফাংশন বুড়ো।

① কেজুন্ট ফাংশন stdio.h নামক হেডুন ফাইল বুড়ো।

② প্রোগ্রাম হেডুন ফাইল এবং নাম স্লিপার #include  
দিয়ে অন্তর্ভুক্ত করতে হব।

ବ୍ୟାକ୍ସନ୍ - ବ୍ୟାକ୍ସନ୍ <= ୧ - ୨୫୫

(6.2)

- ୮୯ ୧-୮

⑧ int = integer ଏକଟି keyword.

ବ୍ୟାକ୍ସନ୍ ଏହା ନିର୍ଦ୍ଦେଖ ମଧ୍ୟ ଏକ ପ୍ରାଗାମ  
main() ଫଂଶନ ଥିଲେ କ୍ଷେତ୍ର ଛାଇ । ତାହା ପ୍ରାଗାମ  
ଏ ଫଂଶନ ଅବଳାଙ୍କି ଲିଖିତ ଛାଇ ।

⑨ Main() ଫଂଶନରେ କର୍ଯ୍ୟକ୍ରମ କ୍ଷେତ୍ର ଉପରେ ମଧ୍ୟ  
(3)

⑩ Return ଏକ ଆଣ୍ଟିକ୍ କ୍ଷେତ୍ର keyword

stdio = standard input

stdio.h = standard input output header

⑪ Print.f ଏକଟି ଲାଇ୍‌ଟ୍ରେଣ୍ଟି ଫଂଶନ ଯାହା ଏକ output

ଦେବା ।

\* main() ଫଂଶନରେ ⑫ ଏକ କ୍ଷେତ୍ର ଲାଇ୍‌ଟ୍ରେଣ୍ଟି ⑬ \*

କ୍ଷେତ୍ର ଲାଇ୍‌ଟ୍ରେଣ୍ଟି

ଏକ କ୍ଷେତ୍ର ଲାଇ୍‌ଟ୍ରେଣ୍ଟି କ୍ଷେତ୍ର ଲାଇ୍‌ଟ୍ରେଣ୍ଟି

ଏକ କ୍ଷେତ୍ର ଲାଇ୍‌ଟ୍ରେଣ୍ଟି କ୍ଷେତ୍ର ଲାଇ୍‌ଟ୍ରେଣ୍ଟି

(କ୍ଷେତ୍ର ଲାଇ୍‌ଟ୍ରେଣ୍ଟି କ୍ଷେତ୍ର ଲାଇ୍‌ଟ୍ରେଣ୍ଟି

# include <stdio.h>

int main()

{

printf("This is our first program");

return 0;

}

Return 0 এর পরিবর্তে getch() ব্যবহার  
করলে ক্ষেত্র অন্তর্ভুক্ত দেখা হবে।

Return 0 এর পরিবর্তে getch() ব্যবহার  
করলে ক্ষেত্র অন্তর্ভুক্ত দেখা হবে।

Q

stdio.h } Some Header files

conio.h

ctype.h

Math.h

stdlib.h

string.h

Time.h

C-9

5.11

## comments & Escape sequence.

comment - এলা যার ক্ষেত্রে লিখাটু কর্মসূচি  
নিশ্চিহ্ন বা অন্যক্ষেত্রে তা একটো লেটে রেজিস্ট্রি  
কর্তৃ কর্তৃ কর্তৃ।

১. অক্ষরাদ।

① single line

② multiple line

i) \*single line comment আঞ্চামত দেখেন

মানুষানু ব্যবহার করা ভাব।

\* শুরু হওয়া // দিয়ে।

\* // দিলে computer পড়িয়ে থাক।

ii) Multiple line comment program রে দেখেন

মানুষানু ব্যবহার করা ভাব।

\* শুরু হওয়া /\* দিয়ে।

\* কোর হওয়া \*/ দিয়ে

Exam.

/ \*  
-----  
-----  
-----  
-----  
-----  
-----  
----- /

④ /\* ~~6-soft/\* and following //~~ \*/  
(Syntax error)

→ multiple line comment use ~~/\* \*/~~  
not ~~1~~

④ Single line comment was created by C++  
Its use in C programming is technically  
invalid, but some compilers will accept it.

### ④ Escape Sequence

C-Program এ কিছি Backslash Characters

\?	→ Question Mark
\t	→ Horizontal space
\v	→ vertical space/ tab
\0	→ null characters
'	→ single quote
"	→ Double "
\\"	→ Backslash

ব্যাখ্যা রচনা হয়ে থা আউটপুট প্রক্রিয়া  
মাত কো

।।a - (Alert) Bell  
।।b - Back space  
।।f - Form Feed  
।।n - New Line  
।।r - carriage Return

Keyword, variable, Data type - 2

୧ ଏହି Token -

- ① keyword → int, float, while
  - ② constants → अंगत value
  - ③ identifier → sum, main
  - ④ strings → "ABC", "Hello"
  - ⑤ special symbol → \*, /, ( ), [ ], { }
  - ⑥ operators → +, -, \*, /

i) Keyword: ଏମନ କିଛି Reserved word ବା ପ୍ରଦ୍ରମ୍ଶିତ

ଜୀବ ମା ଏଣ୍ଡି ନିର୍ଦ୍ଦିଷ୍ଟ ଅର୍ଥ-କୂଳ ଗ୍ରହ ଏଣ୍ଡି

ପ୍ରତ୍ୟାମ ଏଣାଟି ବିନିଷ୍ଟ ହୁଏଥାଏଇନ କରେ ।

⇒ C-program in 32 bit keyword (ANSI)

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

## \* Keyword ব্যবহারের নিয়ম ।

- ① " এখনো চলকের (variable) নাম ইমের ফর্ম্যাট হতে না
- ② প্রোগে এর ছোট শাতে ।
- ③ দুই keyword সাথাপাকি রাখলে মাঝে space  
- ফর্ম্যাট ২৮

long double (valid)

longdouble (Invalid)

## \* Keyword রূপ

## \* variable এ চলক :

program এ- গোল একটি নাম দিয়ে যদি তাৰ

অধীনে Data বৃাধা - হয়, তখন এই চলকের চলক  
বলো ।

## \* Data type - 4 types.

integer - 10, 20 - - - পূর্ণসংখ্যা - 4 bytes

float - দশমিক - 4 "

double - " বড় সংখ্যা - 8 "

char - (A, B, C, - - - \$ - - - ) - 1 bytes

>Data type → 4 types →

- ① char - 1 byte - a, b, c — % C
- ② int - 4 " — ইন্টিজার
- ③ float - 4 " — ফ্লোট (দশমিক) — % f
- ④ double - 8 " — দ্বি দশমিক (15 ঘণ্টা) — % lf

variable এর নাম সংজ্ঞা declaration

" মানের value শুধুক �initialization

variable এর নামের মার্ক (A-Z, a-z),

(0, 1, ..., 9), আন্তরিমকার (-), # ও এস্টেলি

ব্যবহার করা যায়,

role-number ✓

"Anis" ✗

Id1 ✓

anis@ ✗

subject 102 ✓

# anis ✗

\$ anis ✓

② variable ଏହୁ ନାମ digit ଦିଲ୍ଲୀ ଜ୍ଞାତ ହାତେ ଥାଏବେ ନା ।

1 number X

number1 ✓

number1 ✓

a2b ✓

96times X

③ କୋଣ keyword, ସାମଗ୍ରୀର variable ଏହୁ ନାମ  
ଶ୍ରୀମତେ ବାକ୍ୟରେ କହୁ ପାଇଁ ନା ।

float X

Float ✓

math X

Main ✓

for X

④ variable ରୁ ମାତ୍ର ଫଳକ ଘଟନାରେ ଥାଏବେ ନା ।

my\_1st\_variable X

my\_1st\_variable ✓

③ variable ଏହି ତାମାକରଣ ମାର୍ଗରେ ୩୨ character  
ବାକୀର୍ଥ କଣ ଥାଏ, ତାପାରେ character ଏହାରେ  
ବଣାଇନ ।

④ variable /ରୁ Function /Array ଏହି ନିୟମ—  
⇒ (variable ଏହି ନିୟମମଧ୍ୟ)

Q:

X committee

ଯେଉଁ ଏହି ନାମ କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା

ଯେଉଁ ନାମ କିମ୍ବା କିମ୍ବା କିମ୍ବା

X fed

~ fed

X nited

~ nited

X gari

~ gari

ଯେଉଁ ନାମ କିମ୍ବା କିମ୍ବା କିମ୍ବା କିମ୍ବା

Q:

X soldiers +21\_gm

~ soldiers\_+21\_gm

5.19

## keyword, variable, datatype → 2

④ `scanf` →

~~#include <stdio.h>~~

~~int main()~~

{

int number;

printf("Please enter a number");

scanf("%d", &number);

printf("You have press : %d\n", number);

return 0;

}

}

$BC = 5, CD = A \leftarrow \text{High}$

$DE = 5, FE = C \leftarrow \text{High}$

$GP = 0 \leftarrow \text{Initial}$

## ~~upper case to lower case~~

~~Capital~~

← 7mod2

↳ lower case later to upper case later  
without library function.

#include <stdio.h>

```
int main()
{
    printf("Enter a lowercase letter : ");
    scanf("%c", &lower);
    printf("The uppercase letter is = %c",
           lower - 32);
    return 0;
}
```

Ascii → A = 65 , Z = 90

Ascii → a = 97 z = 122

Ascii → O = 48

## Upper to lower

lower -- upper -- with library --

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

```
int main()
```

```
{
```

```
char lower, upper;
```

```
printf("Enter a lower case letter = ");
```

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

```
upper = toupper(lower);
```

```
printf("The upper case letter is = %c", upper);
```

```
return 0;
```

```
}
```

Q

- \* %d, %i - Print any integer - `printf("%d", 5) - 5`
- \* %x, %o - " " " in Hex or Octal format -  
print "%x" 255 - ff

decimal to octal  $\rightarrow$

```
#include <stdio.h>
int main()
{
    int N;
    printf ("Enter a decimal = ");
    scanf ("%d", &N);
    printf ("The octal number = %o", N);
    return 0;
}
```

### ③ অপারেটর:

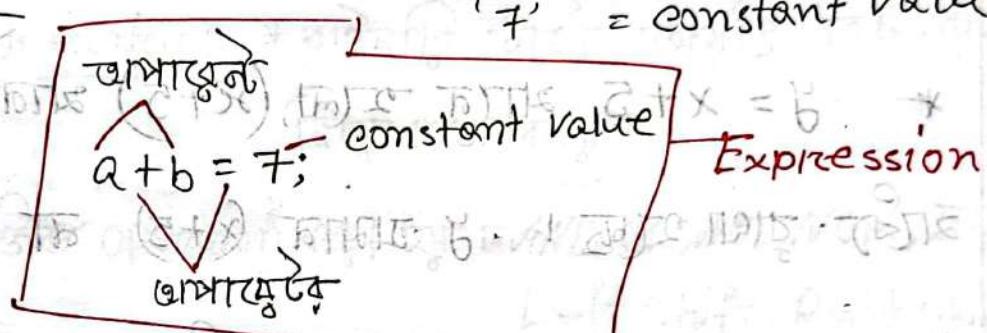
e- আধাৰ্য গণিতিক প্ৰয়োজনীক কাজ নিয়ন্ত্ৰণ কৰাবলৈ এবং বৃত্তজুলৈ বিশেষ চিন্হ (+, -, \*, /) ব্যৱহাৰ কৰা হৈতে, এদেশৰ  
অপারেটোৱ বলে।

$$(a+b=f) \quad ('+', '=) \text{ অপারেটোৱ}$$

$a, b = \text{অপারেটোৱ}$

$f = \text{constant value}$

### ② অপারেটন্ট:



### ④ অপারেটোৱ ধারণা নিয়ে এতে গৃহিৎ তাৰ্ক অপারেটন্ট

田 operators দৰিদ্ৰ অকাৰ,

### ① Arithmetic Operators

+ → যোগাযোগ  $\rightarrow x = 15 + 6 ; n = 21$

- → বিযোগাযোগ  $\rightarrow x = 15 - 6 ; n = 9$

\* → গুণ  $\rightarrow x = 15 \times 9 ; n = 90$

/ → ভাগাযোগ  $\rightarrow x = 15 / 9 ; n = 2$

% → গুচ্ছাযোগ  $\rightarrow x = 15 \% ; n = 3$

## ② Assignment Operator:

=  $y = x + 5;$

+=  $y += 5; \quad y = y + 5;$

-=  $y -= 5; \quad y = y - 5;$

\*=  $x * = 5; \quad x = x * 5;$

/=  $x / = 5; \quad x = x / 5;$

%=  $x \% = 5; \quad x = x \% 5;$

\*  $y = x + 5$  માટે એલા,  $(x+5)$  શાન્દુર રૂપે

અણી રૂધી એલા એન્ટ્યુ 1. ૫ માટે  $(x+5)$  ના।

$y = x$  એલા  $x$  એવું માન રૂપે અણી તુફાની।

## ③ Unary operators:

+	unary plus
-	unary minus
++	<del>increment</del>
--	Decrement

## ④ Relational Operators:

>	Greater than
<	Smaller than
>=	Greater or Equal
<=	Smaller or
==	equal
!=	not equal

- ৫) Logical Operators:  $\&\&$  = Logical AND (এবং)  
 $||$  = Logical OR (অথবা)  
 $!$  = Logical NOT

৬) conditional Operators:

$(a > b)? \quad a : b$

- ৭) Bitwise Operators: \* বাইনারি তারী অ্যাস বিট, এবং  
\* নিচে গত মডেল

\* বিভিন্ন পোজিশন operation করা হয়। AND, OR, NOT, NOR, XOR,  
Left shift, Right shift...

\* can only be used on integers, don't work with float.

& - Bitwise And (এবন)

$>>$  - Right Shift

| - Bitwise OR (মোস)

$<<$  - Left Shift

~ - Bitwise XOR (

~ - Bitwise Not

৮) SPECIAL OPERATOR:

## Bitwise Operators:

Bitwise XOR  $\rightarrow$  Same as  $a \oplus b$

" " as " " 1

$$a = \boxed{0 \ 1 \ 0 \ 0}$$

$$b = \boxed{0 \ 0 \ 1 \ 0}$$

$$a \oplus b = \boxed{0 \ 1 \ 1 \ 0}$$

Bitwise '&'  $\rightarrow$  ~~Same as  $(a + b)$  but~~  $(a * b)$

Bitwise '!'  $\rightarrow$  ~~Same as~~  $(a + b)$

operator topicoperator topic\* Logical Operators:

L : Logical

&& → Logical AND

|| → Logical OR

! → Logical Not

a	b	a&&b	a  b
0	0	0	0
non 0	0	0	1
0	non 0	0	1
non 0	non 0	1	1

\* In case of || or !, if first one is 'non 0' then compiler doesn't complete ahead.

① ~~#include <stdio.h>~~

Output : 12

```
int main()
{
    int a = 10, b = 2;
    b = a++ + 2;
    printf("%d %d", b);
    return 0;
}
```

a = 11  
b = 12

using main() function

.1.

Ope

② ~~#include <stdio.h>~~

```

int main ()
{
    for (int i = 1; i <= 10; i++)
        if (i < 14)
            printf("%d", i);
}

```

Output: 1

(+) '1' because 10 is smaller than 14 ( $10 < 14$ ). So the relation  $a = 10 < 14$  is true. That's why output shows '1'.

③

```

int a;
a = 10 && 9;
printf ("%d\n", a);
}

```

$$\boxed{\begin{array}{l} a = 10 \\ \& a = 9 \end{array}}$$

Show's '0'

'1'

Because if we multiply 2 non zero then the result comes '1' and '1' means true. And ~~true~~ if true then print '1'.

④ {

int a, b = 5;

a = 0 || ++b; // a = 0 || 6, b = 6

printf ("a = %d\n b = %d", a, b)

}

goal in

Output!

a = 1

b = 6

✓ Loop

int main()

{

int i;  
i = 1;

printf ("%d\n", i);

#include <stdio.h>

int main()

int i = 1;

print:

printf ("%d\n", i);

i +=

if (i < 5)

goto print; ] calling the level

Level

STOP

}

## ④ Math Topic (Operators)

### \* Absolute Value (বিপুর্ণ মান)

library function  $\rightarrow$  abs

abs প্রু মাটি- কেন মণ্ডা তাণ্ডল গঢ় জন্মমান পার

$$\text{abs}(-5) = 5$$

$$\text{abs}(5) = 5$$

5

int number = abs(-5);

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

3

5 !

(M95) 050

Johnston

Robin Hood = 21

$$IAN = 62P$$

$$\text{Dose} \times \text{Time} = \text{MSL}$$

OP

## \* square root

\* यह उन double type data type के लिए है।

{

double number = sqrt(25);

printf ("%f", number);

}

\* यह कोटि library function ( $\rightarrow$  sqrt)

## Powers

\* library function  $\rightarrow$  pow \* double datatype

\*  $5^2 \rightarrow \text{pow}(5, 2)$ .

{

double number = pow(5, 3);

,

printf ("%f", number);

০৮

০৯

\* `abs()` এই library function এর জন্য।  
`sqrt()` Math.h header file এর উপরে  
`pow()` এ আছে।

more functions.

→ <code>log()</code>	- <code>sin()</code>
- <code>log()</code>	- <code>cos()</code>
- <code>exp()</code> $\leftarrow e^x$	- <code>tan()</code>

- `round()`
- `truncatel()`
- `ceil()`
- `floor()`

এগুলোতে মান library

function ব্যবহার করে

যে এবং

এগুলো হচ্ছে double

data type use করে।

OPlog {

double  $x = 10.5$ ; //  $\log(10.5)$  এর মান কে  
কষ্ট

double num = log(x);

printf("%lf", num);

}

2.351375

exp

{

double  $x = 1$ ;

double num = exp(x);

exp = 2.718

}

{

(num, "%lf", )

exp(1) =  $e^1$ 

= 2.718...

sin

{

double  $x = 0.35$ ;

double num = sin(x);

printf("%lf", num);

}

0.3428...

Q8

Q8

~~log double log =~~

Q8

{  
double num = log (10.5);

printf ("%lf", num);

}

~~Ans.~~

round 1

#include <stdio.h>

#include <math.h>

int main () {

double x = 5.35;

double num = round (x);

printf ("%lf", num);

}

: 5.0 = x

5.000000

: (x)num = num

: (num, "%lf")

OP

$$\text{round}(5.5) = 6.00$$

$$\text{round}(5.2) = 5.00$$

$$" (0.325) = 0$$

$$" (0.522) = 1$$

round এর কাজ

দ্বিমিকভাবে প্রক্রিয়াজ্ঞান  
করে।

• 5.2 এ 1 মাত্রে

• 5 এর মতো ক্ষেত্রে

ক্ষেত্রে মেমো শর্করা।

## Trunc

: এর কাজ ইলা দ্বিমিক পদ্ধতি (অথবা দ্বিমিকভাবে  
পর্যন্ত অন্তর্ভুক্ত করে)।

```
{ double x = 10.5389;
```

10

```
double num = trunc(x);
```

0

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

```
}
```

trunc 6.596 = 6.000

trunc 0.012 = 0

Project box

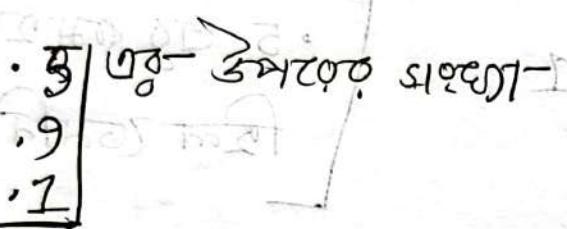
OP

ceil (বাড়ি হাদ)

$$\text{ceil}(2.5) = 3$$

$$", (2.9) = 3$$

$$", (2.1) = 3$$



floor (বাড়ি ঘরে)

$$\text{floor}(2.5) = 2$$

$$", (2.9) = 2$$

$$", (2.1) = 2$$

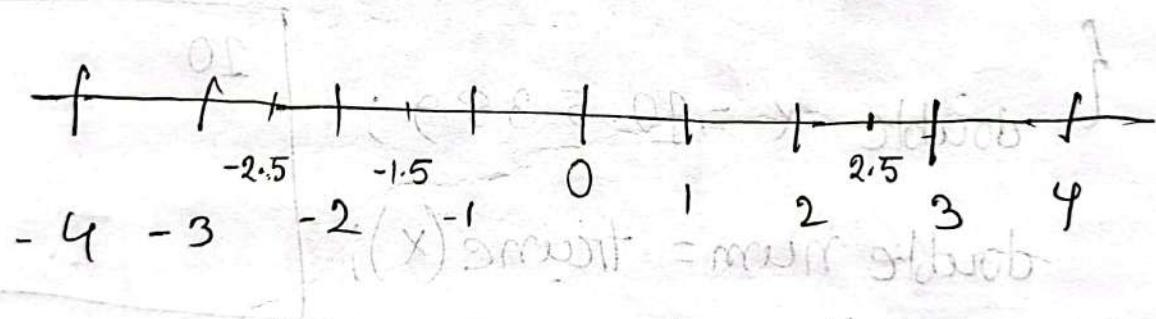
- গ্রুপ নিচে পড়লো

$$\text{floor}(-2.5) = -3$$

$$\text{ceil}(-2.5) = -2$$

-2.5 এর নিচে গ্রুপ হলো 3।

-2.5 এর সমতো পড়লো 2।



So, ceil 2 লা মা বাবুর গুরু বড়ো।

floor = 1 র মাঝে ছোটো।

0 = 0 র মাঝে

End topic

## Arithmetic Operators এবং ব্যবহার,

ক্ষেত্রে জ্ঞানিক পর্যন্ত অন্তর্ভুক্ত ও প্রয়োগ করা হলো

বৃপ্তিকৃত -  $x = 11$

$$x = a^2 - 2ab + b^2 \rightarrow x = a^*a - 2*a*b + b^*b$$

$$Y = AB^2 + C \rightarrow Y = A^*B^*B + C$$

$$X = Y \rightarrow X = Y$$

$$D = X \% Y$$

$$D = \text{sqrt}((b^*b - 4*a*c)/2*a)$$

$$D = \sqrt{(b^2 - 4*a*c)/2*a}$$

~~প্রাপ্তি করা হলো।~~

~~বিনামুক করা হলো।~~

$$X = 4/2 + 8 * 4 - (5+2) \% 3;$$

$$X = ?$$

~~বিনামুক করা হলো।~~

Operators

Priority

( )  $\rightarrow$  1st

$\boxed{* / \%}$   $\rightarrow$  2nd

$+ - \rightarrow$  3rd

$= \rightarrow$  4th

&&

!!

?:

=

$\leq \geq$

$< >$

$\neq$

$\approx$

$\sim$

$\sim$

$\sim$

$\sim$

$$x = 412 + 8 * 4 - (5+2) \% 3$$

$$\text{pass1: } x = 412 + 8 * 4 - (5+2) \% 3$$

$$\text{pass2: } x = 412 + 8 * 4 - 7 \% 3$$

$$\text{pass3: } x = 2 + 8 * 4 - 7 \% 3$$

$$\text{pass4: } x = 2 + 32 - 7 \% 3$$

$$\text{pass5: } x = (2 + 32 - 1 \text{ out of } 3) \% 3$$

$$\text{pass6: } x = (34 - 1 \% 3)$$

$$\text{pass7: } x = 33$$

5.19

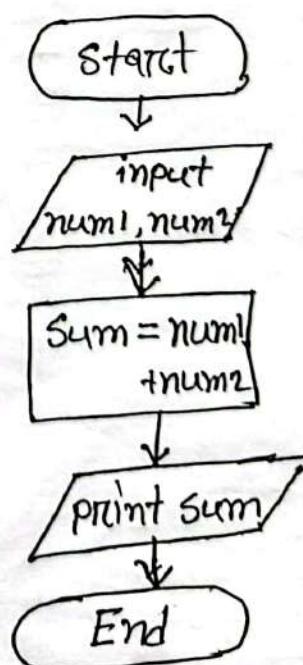
Q) ২টি অন্ধাৰু যোগফল- নির্ণয়ৰ জন্য A, F, C program

কৈতু কষ্ট।

### Algorithm

- ① শুরু,
- ② অন্ধাৰু দুইটি ইনপুট গৈ,
- ③ অন্ধাৰু দুইটি যোগ কৈ,
- ④ পোগফল অদৰ্শন- বাইচি।
- ⑤ কেস।

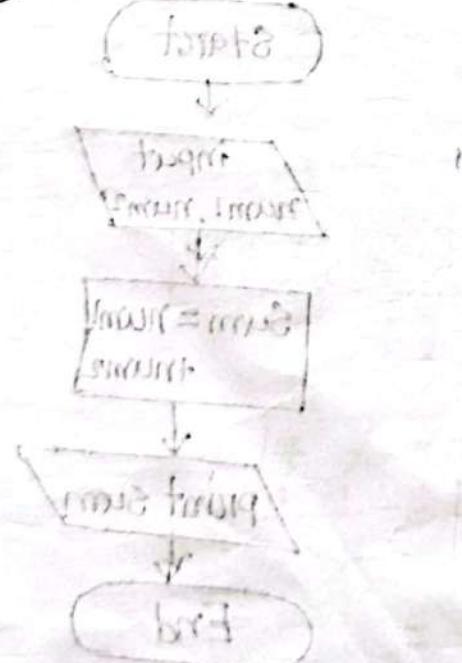
### Flow Chart



### C-Program:

```

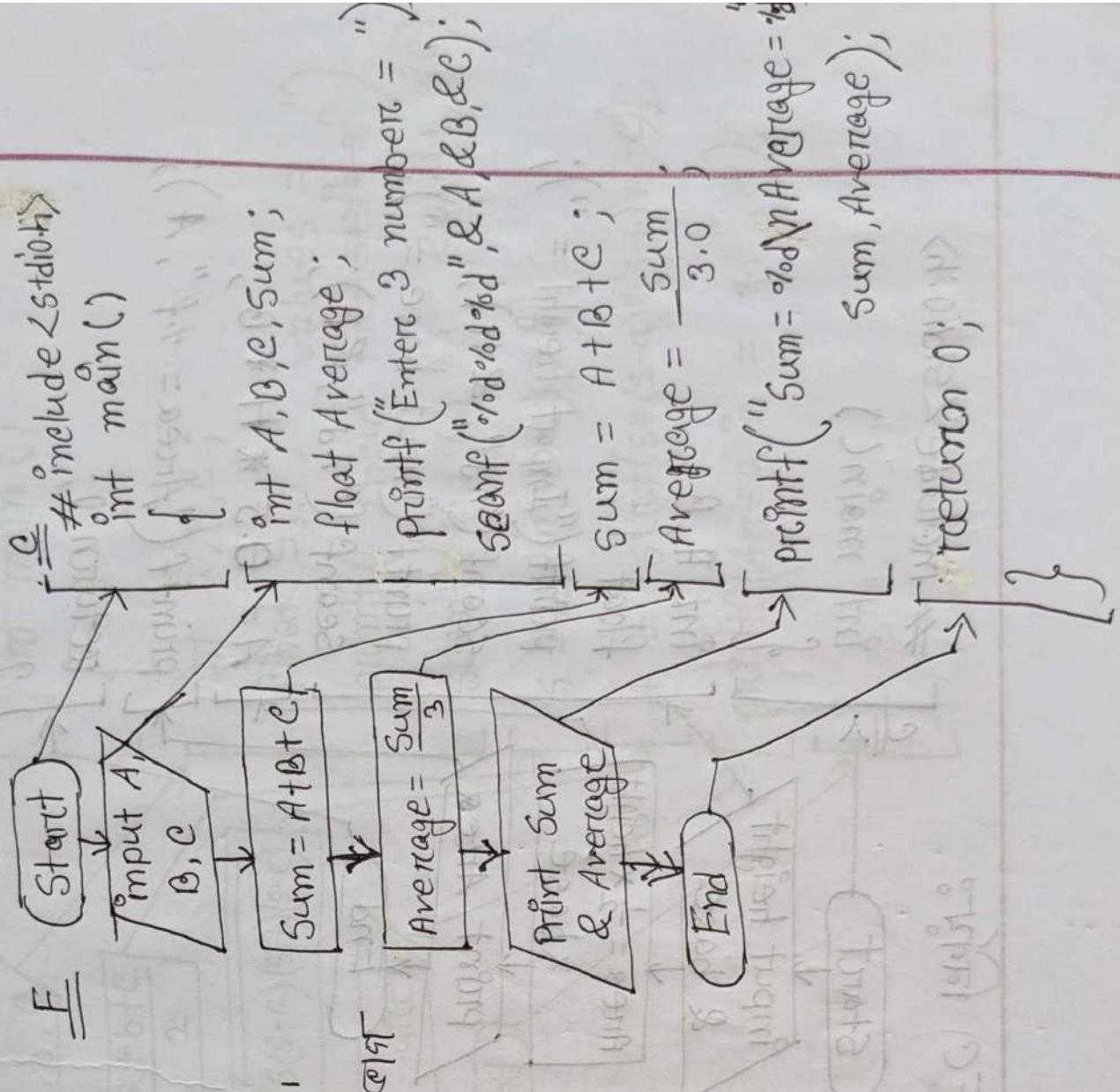
#include <stdio.h>
1 { int main()
2 {
3     int num1, num2; sum;
4     printf("Enter two numbers = ");
5     scanf("%d %d", &num1, &num2);
6     sum = num1 + num2;
7     printf("The sum is = %d", sum);
8     return 0;
9 }
```



4) A  
 ① Start  
 ② Input A, B, C  
 ③ Sum = A + B + C  
 ④ Average = Sum / 3  
 ⑤ Print Sum & Average  
 ⑥ End

5) A, F, C Output -

(5-31)



৫) প্রোগ্রাম একটি পরিপূর্ণ তরঙ্গে A,F,C নথিতে

A

① সুব্রত-

২) বেজের অন্তর্ভুক্ত করা হয়।

৩) ক্ষেত্রফল =  $\frac{1}{2} \times \text{ভূমি} \times \text{উচ্চতা}$   
পৃষ্ঠা বরবরা দ্বারা ক্ষেত্রফল  
পৃষ্ঠা করা হয়।

৪) ক্ষেত্রফল প্রদর্শন করা

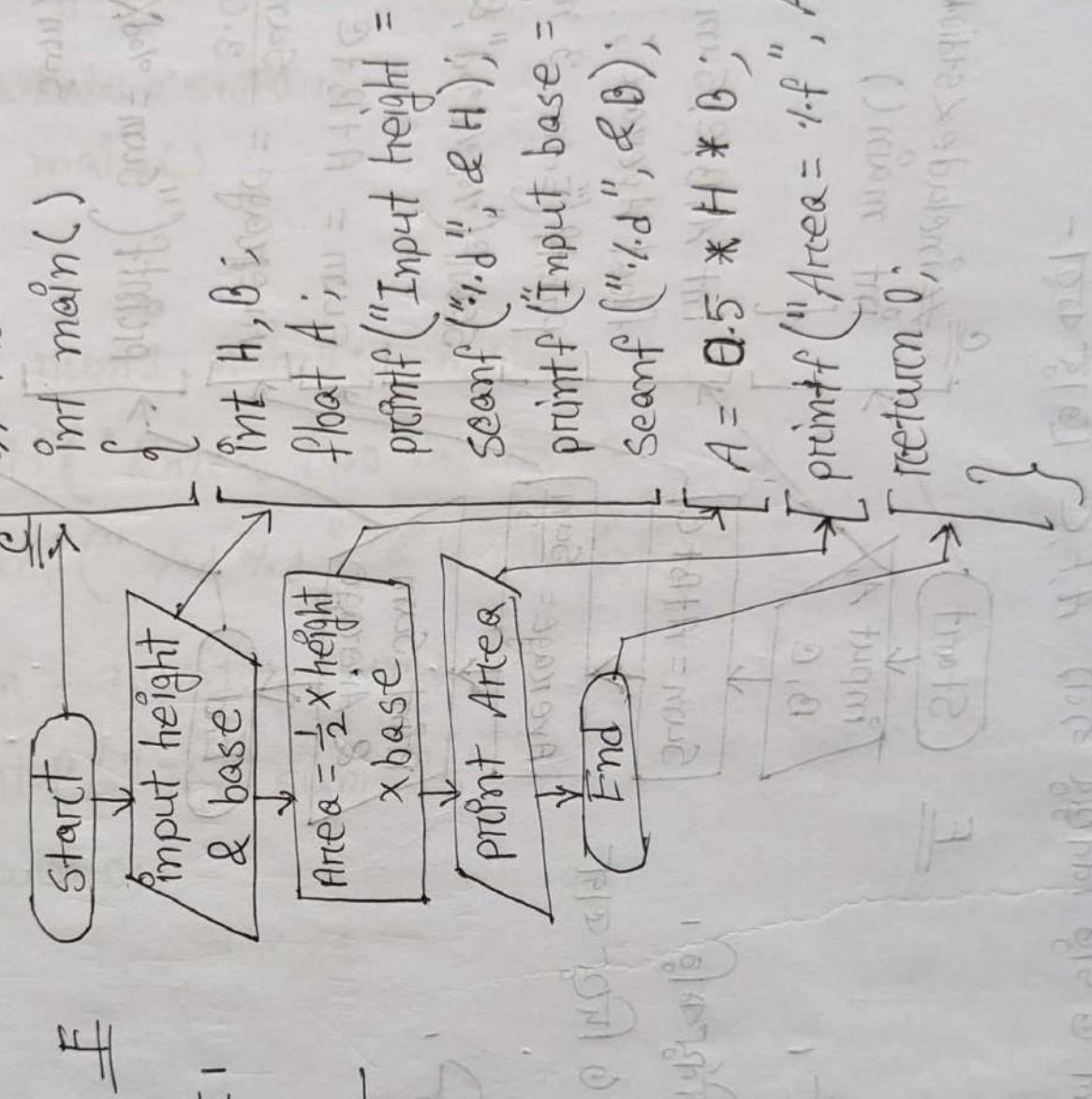
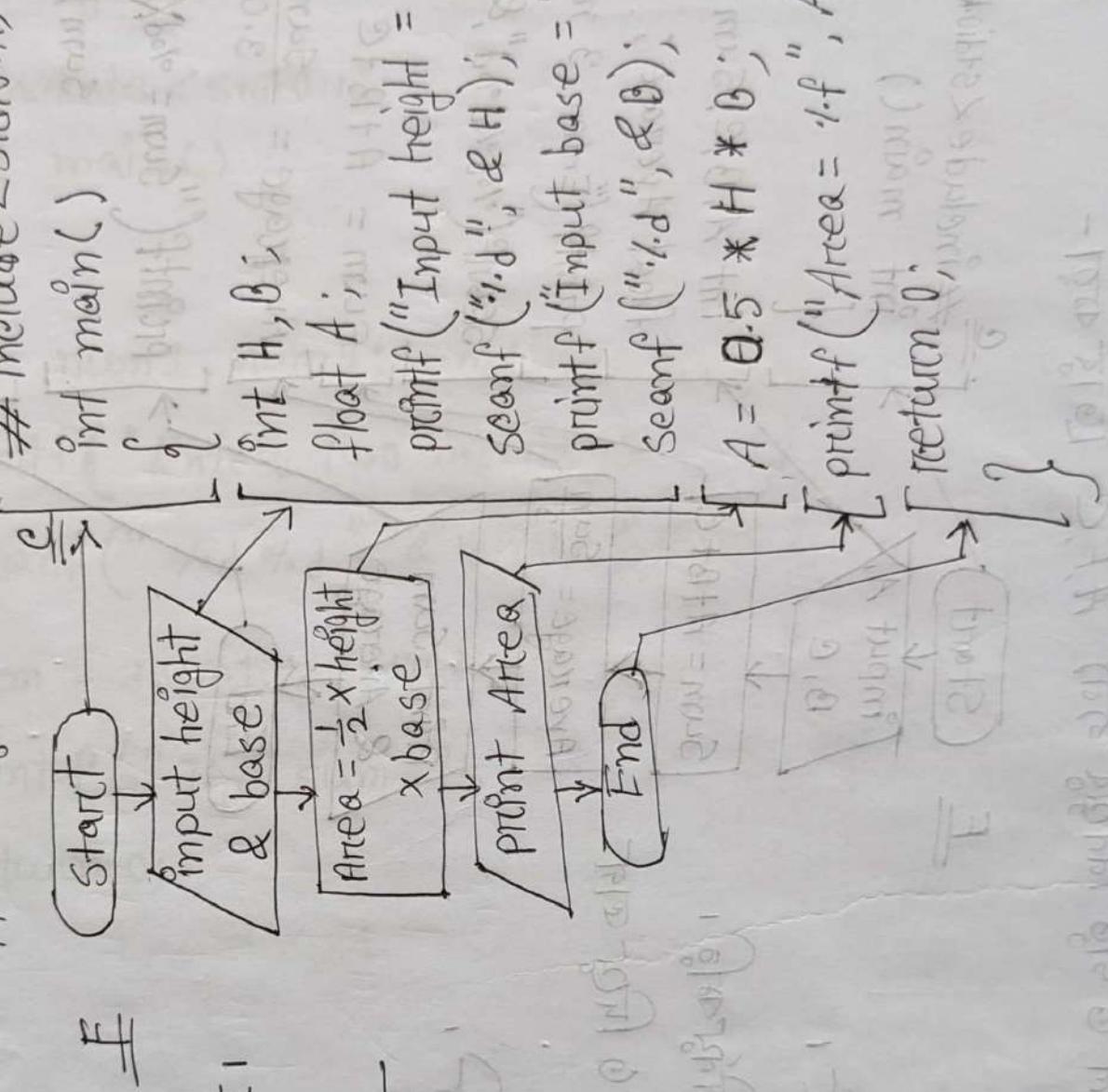
৫) কোড-

১) প্রক্রিয়া প্রস্তুত করা

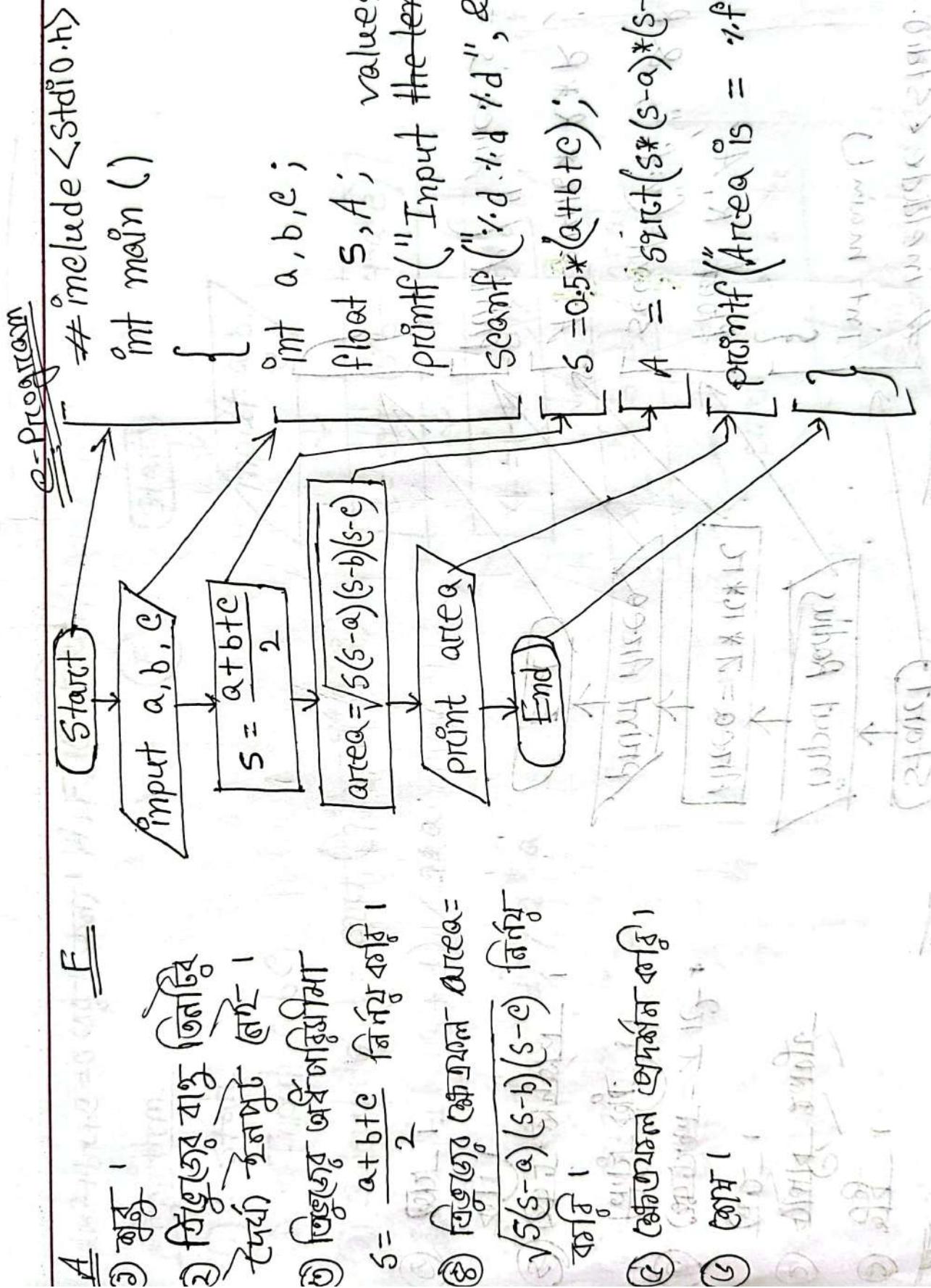
২) প্রক্রিয়া প্রস্তুত করা

৩) প্রক্রিয়া প্রস্তুত করা

৪) প্রক্রিয়া প্রস্তুত করা



4) କ୍ଷେତ୍ରଫଳ ନିର୍ଣ୍ଣୟ କରିବାର ପଦ୍ଧତି ଆଜାନ୍ତର ଏକ ପ୍ରଗରାମ ଲିଖିବାରୁ A, F, C - program ଲିଖିବାରୁ



ବିଶ୍ୱାସ କୋର୍ପ୍ସନ ନିର୍ଦ୍ଦେଶ A, F, C

4

② ବାଜାର୍ କଣାଖୁଲ  
କଟାଇ

③ କୋଣକୁଣ୍ଡଳ =  $\pi r^2$ ,  
କଣକୁଣ୍ଡଳ କାହାରେ

କଣକୁଣ୍ଡଳ - ପରାମର୍ଶକ

④ କଟାଇ

କଟାଇ କାହାରେ  
କଟାଇ କାହାରେ

କଟାଇ

F

(Start)

Input Radius

$$\text{Area} = \pi * \text{Radius}^2$$

Print Area

କଟାଇ

କଟାଇ

କଟାଇ

କଟାଇ

କଟାଇ

କଟାଇ

E

#include <stdio.h>

int main()

{  
 float R, A;  
 printf("Enter R, A, = ");  
 scanf("%f", &R);  
 A = 3.1416 \* R \* R;  
 printf("Area is %f, A = %f", A);  
}

};

};

};

};

};

};

};

};

};

};

};

};

};

};

};

};

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \quad d = \sqrt{b^2 - 4ac}$$

Quadratic demo

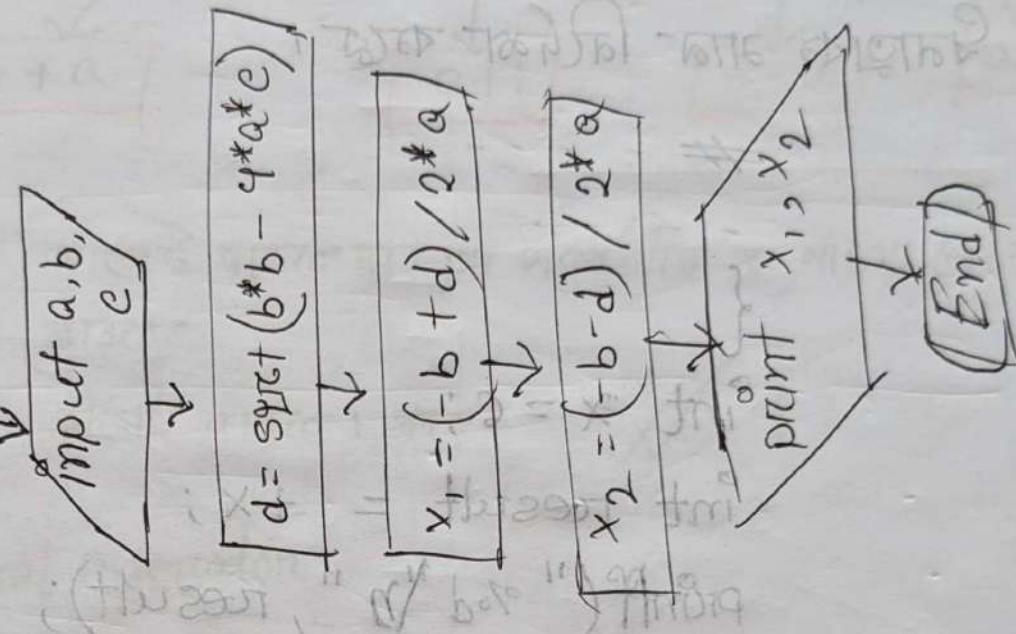
$$ax^2 + bx + c = 0 \text{ (जब अन्य) } A, F, C \text{ फॉर्मूला -}$$

Algorithm.

- ① : Start
- ② : Input a,b,c
- ③ Calculate  $d = \sqrt{b^2 - 4ac}$
- ④  $x_1 = \frac{(-b + d)}{2a}$
- ⑤  $x_2 = \frac{(-b - d)}{2a}$
- ⑥ Print  $x_1, x_2$
- ⑦ End.

(F)

(Start)



C.5.51

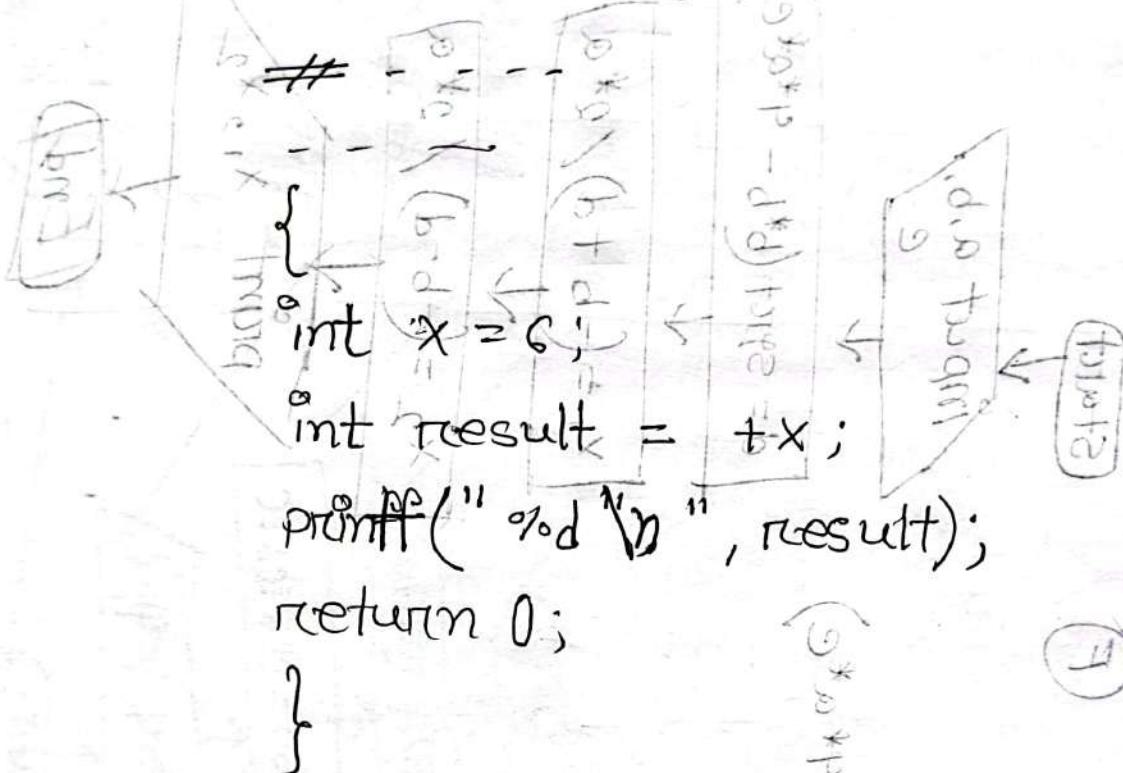
\* Assignment operators

C.5.58

## Unary operators

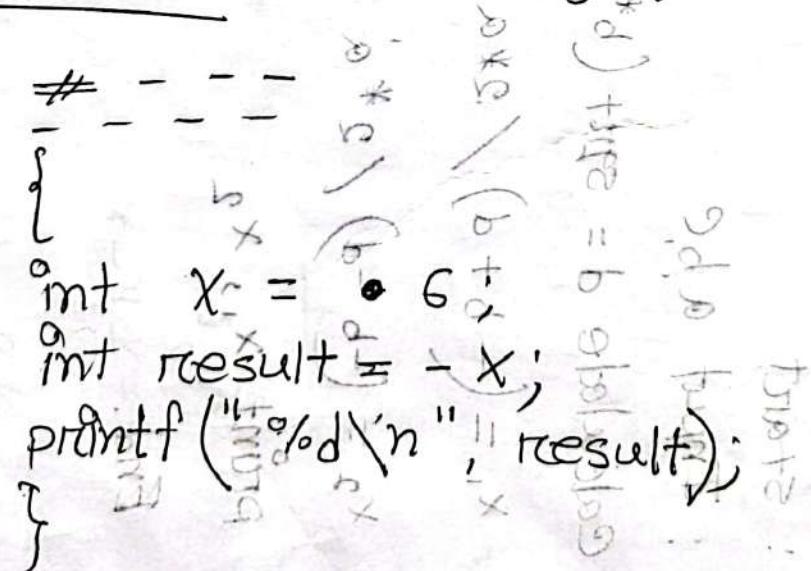
### i) Unary Plus:

ইন্টার্ভ মান- নির্দিষ্ট করছে।



### ii) Unary minus:

মুনাফা মান কুসাই।

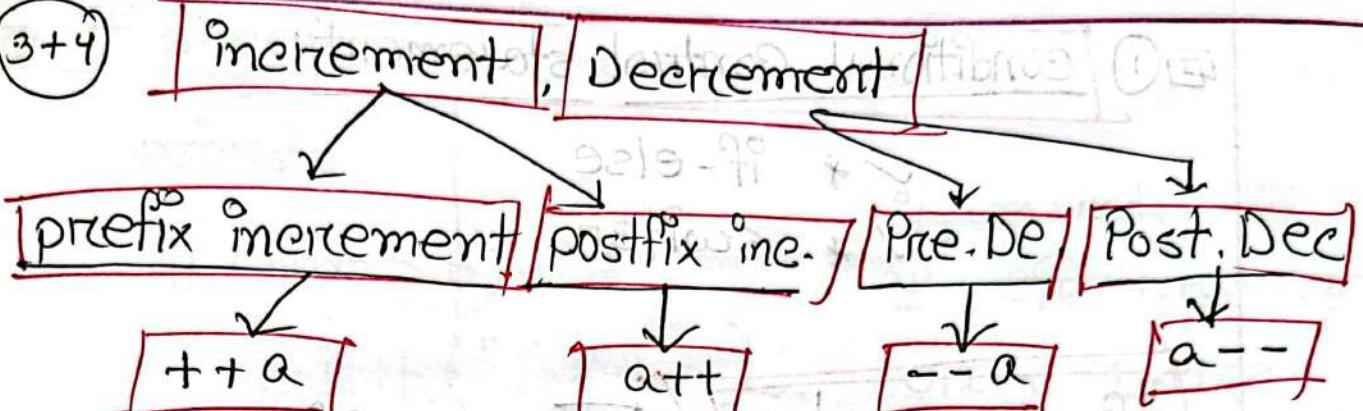


- 4
- 5
- 6
- 7
- 8
- 9
- 10

marking sheet

H.C.G.E.A. (60-75) 0 = 34.6 + 3(6)

3+4



$x++$  = আগের পাতত না। এখন পুরোটো  $x$  আপডেট করে দেয়া

$++x$  = আগেই পাতত না। ~~পুরোটো~~



## Relational Operator

~~#~~ control statement:

- ① conditional control System
- ② Loop control statement

## ① conditional control statement

P+E

✓ \* if-else

✓ \* switch

5.60

\* if-else syntax:

if (condition)

{

Statement 1;

Statement 2;

-----

Statement n;

} for if condition ①

{

Statement 1;

Statement 2;

-----

Statement n;

}

if, else-if অনেকস্থানে

ব্যবহার করা হয়।

else ক্ষেত্রে মাত্র একটি

ব্যবহার করা হয় program এ

শৈলী।

Key points of if-else statement

if condition good ②

else condition

if condition good ③

## if statement:

```
printf  
if (num > 0)  
    printf ("Positive.");
```

if এর উদাহরণ

if যদি ১ হবে তা

জয়লা '1' হবে।

১ মান হলো

## if statement:

```
if (num > 0)  
    printf ("Positive.");
```

if যদি ১ হু গৱল

অ true . True হল

if এর উদাহরণ অ if statement মানবে তা print কো

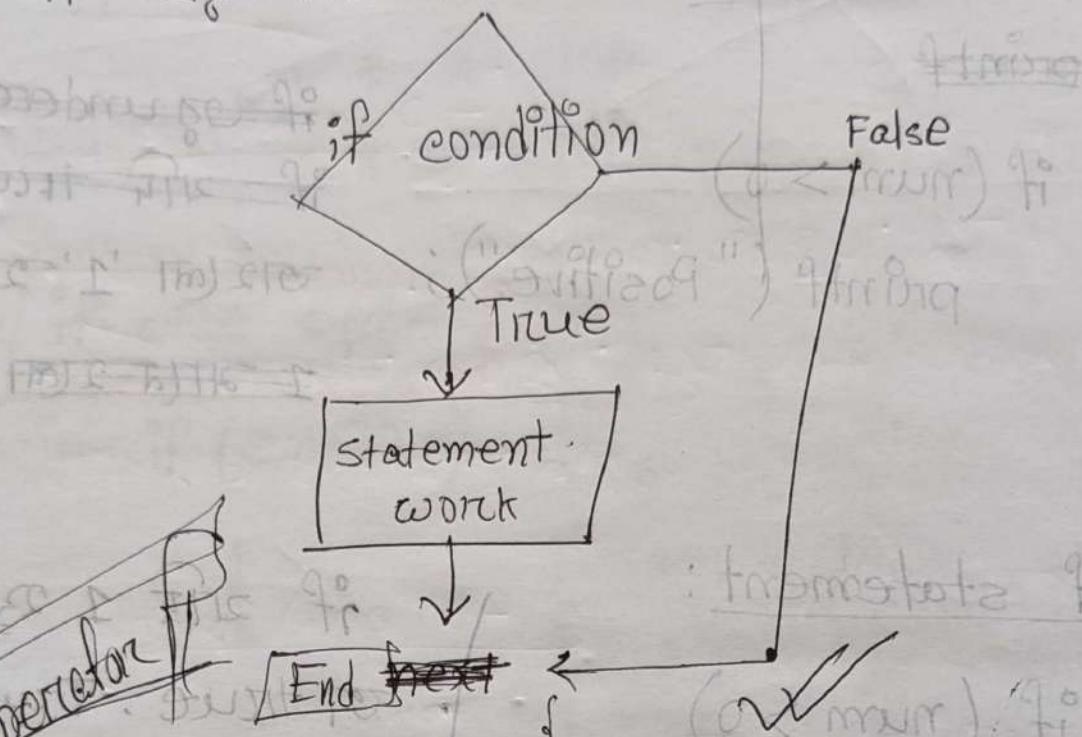
single statement - এল  
bracket লাগিবে না।

or. if (condition)

```
{  
    printf ("")  
}
```

if

if अंडे Flowchart:



~~Logical Operator~~

```
if(ch == 'a')  
printf("vowel");  
else if(ch == 'e')  
---- ("---");  
else if(ch == 'i')  
---- ;  
else if(ch == 'o')  
---- ;  
else if(ch == 'u')  
---- ;  
else  
printf("consonent");
```

if (ch == 'a' || ch == 'e' || ch == 'i'  
|| ch == 'o' || ch == 'u')

printf("vowel");

else  
printf("consonent");

{

y  
do

কোন Year-তে 400 দ্বারা ভাগ করলে অগশ্য '0' রইল  
তা Leap Year.

Leap Year এর একটি C-Program

\* কোন Year যদি 4 দ্বারা ভাগ করলে অগশ্য '0' রয়ে 100  
দ্বারা ভিত্তি ভাগ-কর্তৃ আ-মানুষ তবে তা Leap Year.

Leap Year নির্ণয়ের C-Program

▪ Variable

① Local Variable (Main function এর)  
বাস্তু রাখার জন্য

② Global Variable

#include < stdio.h >

int main()

{

    int a = 10

    local variable

    printf("%d", a);

    display();

}

void display()

# switch case statement

{5.83}

## Variable:

### ① Local Variable -

- Main function এতে মাত্র গঠিত হবে।
- সব ফাংশনের গঠিত হবে যেই ঘোষণা দ্বারা আনা গোপনীয় ঘোষণা করে না।

### ② Global Variable -

- Main function (main()) এতে বা অন্য ফাংশন

এতে ক্ষেত্র ব্যবহার করা হবে।

#include<stdio.h>

• int a = 3;

int b = 4;

int main ()

{

printf ("%d", a+b);

}

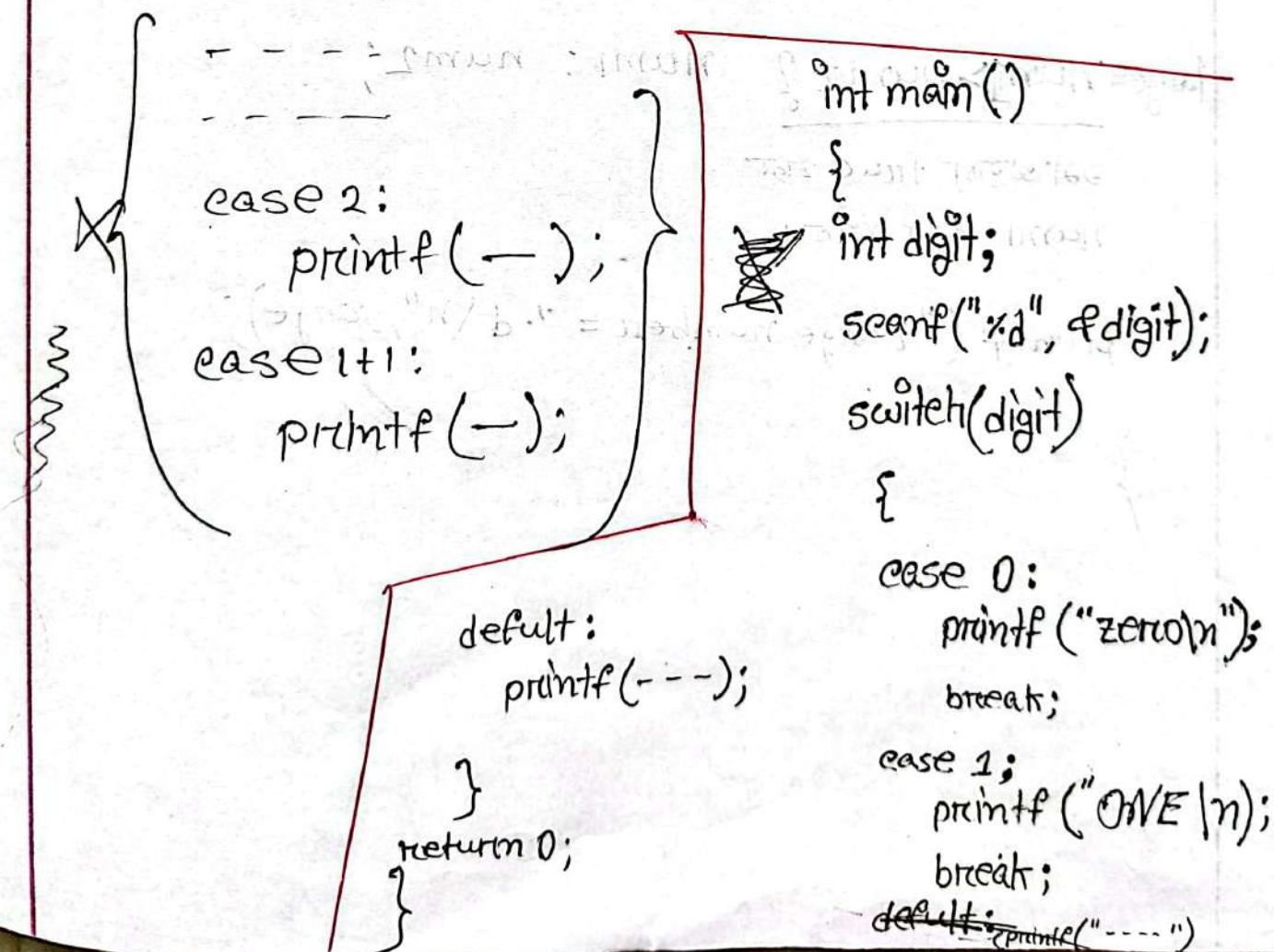
- এর ঘোষণার ব্যবহার হবে।

5.84

## Switch control system

Extra used keyword for switch case →  
switch, case, break, default  
main                                    optional

- \* The expression used in switch must be integral type. (int, char, enum).
- \* default block can be used anywhere. The position of default doesn't matter.  
`default`
- \* Two case labels cannot have same value.



## Let's learn conditional or ternary operators:

\* ഒരു അളവുന്ത് വിധം എങ്കണ്ട്!

\*  $\text{num1} ? \text{num2} : \text{num3}$

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

```
int main() {
```

```
{
```

```
    int num1, num2, large;
```

```
    scanf("%d %d", &num1, &num2);
```

large = num1 > num2 ? num1 : num2;

    true value

    false value

```
    printf("Large number = %d\n", large);
```

```
}
```

```
}
```

: 0

scanf

num1

: 1

scanf

num2

: 2

: Large

number

: 1

number

: 0

LoopLoop -

① for loop

② while

③ do while

- \* for loop →
  - initialization
  - condition
  - increment / decrement

int i;

for (i=1; i&lt;=5; i++)

printf("A");

\* একাবিক statement থাকলে second bracket দিতে হবে।

While loop →

- \* শুধু condition থাকবে, initialization, increment part থাকবে না।

\* আগেই variable initialization রাখতে হবে।

\* পরে increment রাখতে হবে।

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

```
int main()
```

```
{
```

```
int i;
```

```
i = 1;
```

```
while (i <= 10)
```

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

```
} i++;
```

Output:

initialization

condition

- increment/decrement

parts of while loop

1

2

3

4

5

6

7

8

9

10

# Loop

## 4) do while loop -

\* କମ୍ପ୍ୟୁଟର ଦେଖିଲେବୁଛି

\* ନିଚେ କାହାରେ କମ୍ପ୍ୟୁଟର ଦେଖିଲେବୁଛି

\* condition part କାହାରେ କମ୍ପ୍ୟୁଟର ଦେଖିଲେବୁଛି, ଏବଂ ଯେବେଳେ

sequence: - initialisation

condition

- printf

- increment / decrement

- condition

```
{  
    int i;  
}
```

```
i = 1;
```

```
do
```

```
: {
```

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

```
    i++;
```

```
} while(i <= 10);
```

```
return 0;
```

```
}
```

Output

1

2

3

4

5

6

7

8

9

10

1-10 for, for, while, do-while

for  
#include <stdio.h>

```
int main()
{
    int i;
    for (i=1; i<=10; i++)
        printf ("%d\n", i);
    return 0;
}
```

Output

1  
2  
3  
4  
5  
6  
7  
8  
9

(more)

do-while

✓

while

✓

```
int main()
{
    int i=1;
    while (i<=10)
    {
        printf ("%d\n", i);
        i++;
    }
    return 0;
}
```

```
int main()
{
    int i=1;
    do
    {
        printf ("%d\n", i);
        i++;
    } while (i<=10);
    return 0;
}
```

Loop

B3

## Loop

\* Break, Continue → in loop }

break = loop-ରୁ ତରମ୍ଭ କରିଦେବୁ ।

loop ଥିବା କାହାର ସର୍ବକାରୀ କରିଦେବୁ ।

Continue, loop ଓ continue ମଧ୍ୟରେ ଏହି

ନିଚ୍ଚତ୍ଵ ପର୍ଯ୍ୟନ୍ତ କରିବାକୁ ନା ।

\* Code ରୁ କୋଣ କରିବାକୁ ନା ଛାଇଲେ,

catfile () {

- program

}

}

→ (r>i) fi

; f + i

; f = i

; f++

[more]

## Loop

\* ସିନାତ୍ରା ପୂର୍ବ ଅଞ୍ଚଳୀୟ factorial ନିକ୍ଷେତ୍ର:

$$4! = 1 \times 2 \times 3 \times 4$$

$$3! = 1 \times 2 \times 3$$

$$0! = 1$$

\* ST. ଶାଖା ଓ ଲାଭାଗ୍ରହ (GCD/HCF & LCM):

$$\text{num1} = 60$$

$$\text{num2} = 24$$

C-Program ଓ GCD, LCM

ଏହି କ୍ଷେତ୍ର ପ୍ରଥମ GCD

$$\text{ଏହି ଏବେ, } LCM = \frac{n_1 \times n_2}{GCD}$$

ଯୁଦ୍ଧରେ ବାରମ୍ବାର କଢ଼ିତେ ଥିଲା।

$$\underline{\text{GCD}}: n_1 = 60$$

$$n_2 = 24$$

$$\frac{n_1}{60} \quad \frac{n_2}{24}$$

$$60 \% 24 = 12$$

$$24 \% 12 = 0$$

$$12 \% 0 =$$

$$60 = 2^2 \times 3^1 \times 5^1$$

$$24 = 2^3 \times 3^1$$

$$GCD = 2^2 \times 3 = 12$$

$$LCM = 2^3 \times 3 \times 5 = 120$$

$$\text{Or. LCM} = \frac{\text{num1} \times \text{num2}}{\text{GCD}}$$

$$= \frac{60 \times 24}{12} = 120$$

normal process

remainder (ବିଶେଷଜ୍ଞ)

ଯଥିନ 0 ଟଙ୍କା ଆମରେ ଉଦ୍ଦତ  
(କୌଣସି),

$\therefore 12 \text{ Ans.}$

L

Loop

$$\text{LCM} = \frac{60 \times 30}{12} = 120$$

A 60, 30

 $n_1 = 60, n_2 = 30, \text{ remainder}$ 

$$\begin{array}{r} 60 \\ \times 30 \\ \hline 0 \end{array}$$

$$\therefore (\text{GCD} = 30)$$

$$\therefore \text{LCM} = \frac{60 \times 30}{30} = 60$$

#include &lt; stdio.h&gt;

int main() {

int num1, num2, n1, n2, remainder, gcd, lcm;

printf("Enter 2 numb. = ");

scanf("%d %d", &amp;num1, &amp;num2);

n1 = num1  
n2 = num2  
while (n2 != 0) {

n1 % n2 = remainder;

n1 = n2;

n2 = remainder;

}

gcd = n1;

lcm = ~~(num1 \* num2) / GCD;~~

printf("%d %d", gcd, lcm));

return 0;

}

motee

Again loop

\* Sum of digits:

123

431

$$= 1 + 2 + 3 = 6 \text{ or } 4 + 3 + 1 = 8$$

① Sum = 0

num = 123

temp = num

while (temp != 0)

r = temp % 10

Sum = sum + r

temp = temp / 10

$$\boxed{\begin{array}{l} \text{sum} = 0 + 3 \\ \quad \quad \quad \boxed{0251} \end{array}}$$

$$\begin{aligned} &= 3 + 2 \\ &= 5 + 1 \end{aligned}$$

User द्वारा num नियंत्रित

temporary file पर बहार

123 द्वारा आकाश कहुए

$$10 \overline{)123} \quad \boxed{12}$$

$$\begin{array}{r} 10 \\ 23 \\ \hline 20 \end{array}$$

$$\boxed{3} \rightarrow r$$

$$10 \overline{)12} \quad \boxed{1}$$

$$\boxed{2} \rightarrow r$$

जावाखल २ला temporary बना आता

उपर्युक्त " आलादा होता number.1

$$10 \overline{)10} \quad \boxed{1}$$

# Loop

L

{ cool নিয়ে

## C-code

{

int num, temp, sum=0, reminder;

scanf ("%d", &num);

temp = num;

while (temp != 0) // অগমল 0 তা হত্তয়ি

পর্যন্ত loop করো।

{

~~temp =~~ temp % 10; // 10 দিয়ে লেজ করলে অসকে

reminder আলাদা number.

sum = sum + reminder,

temp = temp / 10; // 10 দিয়ে অবশ্যে অগমল

হবে temp.

}

printf ("Sum = %d", sum);

}

L

\* Reverse an integer | 123 → 3 2 1

\* sum of digit & reverse അല്ലെങ്കിൽ sum.

\* 10 ദിശയിൽ എണ്ണം → അനുഭവ temporary.

→ അനുഭവ remainder എണ്ണാദിനും



\* ഗോളിത്വം sum = sum \* 10 + rem

{

i ("sum = sum + rem")

while (temp != 0)

{

rem = temp % 10;

sum = sum + rem + 1

sum = sum \* 10 + rem;

temp = temp / 10;

}

printf ("%d", sum);

123 → 3 2 1

$$\begin{array}{r} 10 \\ \overline{)123} \\ 10 \\ \hline 23 \\ 20 \\ \hline 3 \end{array} \text{--- rem}$$

$$S = 0 \times 10 + 3 = 3$$

$$\begin{array}{r} 10 \\ \overline{)12} \\ 10 \\ \hline 2 \end{array} \text{--- rem}$$

$$S = 3 \times 10 + 2 = 32$$

$$\begin{array}{r} 10 \\ \overline{)1} \\ 0 \\ \hline 1 \end{array} \text{--- rem}$$

$$S = 32 \times 10 + 1 = 321$$

Loop

## \* Palindrome number:

→ reverse करके वहाँ अपि तरीके से उसे बहुत ही आसानी से जान सकते हैं।

After reverse;

if (num == sum)

printf ("Palindrome");

else

printf ("Not Palindrome");

return 0;

}

## \* Armstrong:

$$\begin{aligned} 153 &\rightarrow 1^3 + 5^3 + 3^3 \\ &= 1 + 125 + 27 \\ &= \underline{\underline{153}} \end{aligned}$$

while (temp != 0)

{  
    n = temp % 10;

    sum = sum + n \* n \* n;

    temp = temp / 10;

}

if (num == sum)

printf ("Armstrong");

L

\* count : 123 → 3 ഫോറ്റുകൾ

\* strong number: 123 → 1 + 2 + 3! = 1 + 2 + 6

$$1! + 2! + 3! \leftarrow 2! = 6$$

\* factorial (സംഖ്യയെ പൊലിക്കണ)

```
int num, count = 0;
scanf("%d", &num);
```

```
while (num != 0)
```

```
    num = num / 10;
```

```
    count++;
```

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

```
(num == 0) { i = 0; q = 1; }
```

```
(i + 1) <= n : i = i + 1
```

```
i * fact = fact
```

```
(fact = fact * i)
```

```
fact + num = num
```

```
q = q * fact
```

## \* Strong Number

$$\{ \quad \begin{array}{c} 1 \ 2 \ 3 \rightarrow 1! + 2! + 3! \\ = 9 \end{array} \quad \} \quad X$$

$$\underline{1 \ 4 \ 5} \rightarrow 1! + 4! + 5!$$

$$= 1 + 24 + 120$$

$$= 145$$

Same

## C-Code

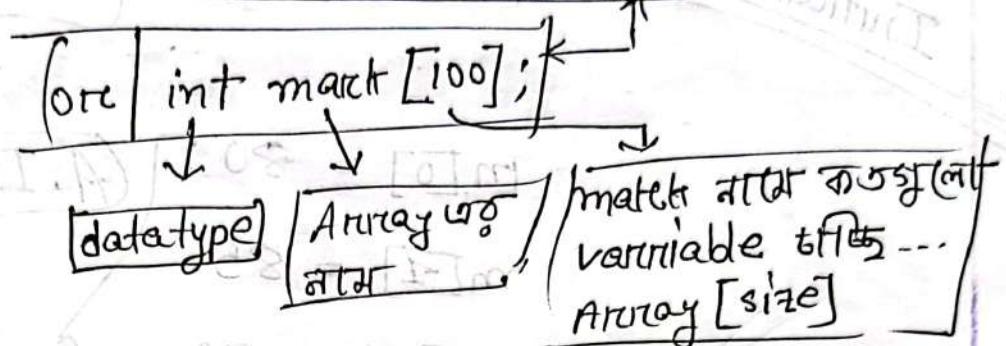
```
{ int num,temp, rc, i, fact=1, sum=0,  
scanf( " %d ", &num );  
temp = num;  
while ( temp != 0 )  
{ rc = temp % 10;  
for ( i=1; i<=rc; i++ )  
fact = fact * i;  
sum = sum + fact;  
temp = temp / 10;  
}
```

}      {      }      {      }      {      }

printf( " %d "  
if ( num == sum )  
printf( " Strong "  
else  
printf( " not strong " )

Array-ArrayArray

[int marks[1], int marks[2] ... int marks[100],]



int Num[500];

float Num[100];

\* Array is a collection of variables of same datatype

■ Array start with '0' from '0' in tri

,, size can't be a negative numb.

int marks[5];

Array size 5

(([E]rr, "ba") first variable. marks[0]) tri

(([E]rr, "ba")) tri

- [1]
- [2]
- [3]
- [4]

Array

A

Array  
Initialization:

int m [5];

(A. Declared)

m[0] = 80

(A. Initiated)

m[1] = 85

m[2] = 90

m[3] = 82

m[4] = 100

↙ int m [5] = { 80, 85, 90, 82, 100 };

↙ int m [ ] = { 80, 85, 90, 82, 100 } )

Print Array:

printf ("%d", m[0]);

printf ("%d", m[1]);

printf ("%d", m[2]);

— — —

[0]  
[1]  
[2]  
[3]

A

Array

```
for (i=0; i<=4; i++)
    printf("%d\n", m[i]);
```

output

80  
85  
90  
82  
100

user input in Array:

```
scanf("%d" & m[0]);
```

```
scanf("%d" & m[1]);
```

```
scanf("%d" & m[2]);
```

```
for (i=0; i<=4; i++)
    scanf("%d" & m[i]);
```

more

## Again Array

```
#include<stdio.h>
int main()
{
    int num[5], i, sum=0, temp;
    printf("Please enter 5 numbers = ");
    for(i=0; i<5; i++)
    {
        scanf("%d", &temp);
        num[i] = temp;
    }
    for(i=0; i<5; i++)
    {
        sum = sum + num[i];
    }
    printf("%.d is sum", sum);
}
```

A

maximum

```
{  
int num[100], n, i;  
printf ("How many numbers: ");  
scanf ("%d", &n);  
for (i=0; i<n; i++)  
    scanf ("%d", &num[i]);  
int max = num[0];  
for (i=1; i<n; i++)  
{  
    if (max < num[i])  
        max = num[i];  
}  
printf ("Max = %d\n", max);  
}
```

A

minimum

```
int min[] = num[0];
```

```
for (i=1; i<n; i++)
```

```
{ if (min > num[i])
```

```
    min = num[i]
```

```
}
```

```
printf("Maximum = %d", min)
```

```
}
```

A

## Fibonacci numbers

{

01123#5813-~

int a[30],n;

$$a[i] = a[i-1] + a[i-2]$$

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

$$a \otimes [0] = a$$

$$\alpha [I] = 1$$

```
for(i=2; i<n; i++)
```

$$a[i] = a[i-1] + a[i-2] \quad // a[5] = a[5-1] + a[5-2] \\ = a[4] + a[3]$$

```
printf("n");
```

$$\begin{array}{r} 3 + 2 \\ = 5 \end{array}$$

```
for(i=0; i<n; i++)
```

```
printf("%d\n", a[i])
```

(return 0)

A

• Statement of problem & most common problems

## Searching numbers in Array :

— linear searching [~~Elements of array are sorted~~ Array size  $2^{24}$ ]

— Binary

{

int num[] = {10, 2, 15, 20, 35, 46, 85};

int value; pos = -1;

printf("Enter the value you want : )

scanf("%d", &value);

for (i=0; i < 7; i++)

if (pos == -1)

printf("Not found")

{ if value = num[i]

else

printf("found & position %d, pos);

{ pos = i; }

return 0;

break;

}

}

(If found "but = open) returning

Copied index from 1 array to another.

{

int array1[4] = {5, 6, 7, 10}; array2[]

~~printf~~

~~array2~~

for (i = 0; i < 4; i++)

array2[i]

{

int

array2[4];

for

{

array2[i] = array1[i];

for

{

printf

{

array2 = %d", array2[i])

A2D - ArrayDeclare

~~int array [row size] [column size]~~

~~int a [3] [4];~~

$\therefore$  num. of elements = row \* col

$$= \underline{\underline{12}}$$

~~int a [3] [4];~~

	0	1	2	3
0	0,0	0,1	0,2	0,3
1	1,0	1,1	1,2	1,3
2	2,0	2,1	2,2	2,3

	0	1	2	3
0	5	6	7	8
1				
2				

~~int a [3] [4];~~

$$a[0,0] = 5$$

$$a[0,1] = 6$$

$$a[0,2] = 7$$

$$a[0,3] = 8$$

## initialization:

[int A[3][4] = {

{5, 6, 7, 8},

{15, 16, 17, 18},

{25, 26, 27, 28}

	0	1	2	3
0	5	6	7	8
1	15	16	17	18
2	25	26	27	28

```

printf("%d", A[0][0]);
for(i=0; i<=2; i++)
{
    for(j=0; j<=3; j++)
    {
        printf("%d", A[i][j]);
    }
}

```

```

printf("\n");
printf("%d", A[1][0]);
printf("%d", A[1][1]);
printf("%d", A[1][2]);
printf("%d", A[1][3]);
}

```

```

}
printf("\n")

```

0	1	2	3
5	6	7	8
15	16	17	18
25	26	27	28

A

Scan

```

int array[3][4], i, j;
for (i=0; i<3; i++) {
    for (j=0; j<4; j++) {
        printf ("array [%d][%d] = %d, (%i,%j)", i, j);
        scanf ("%d", &array[i][j]);
    }
    printf ("\n");
}

```

A

Simple matrix: 2D Array, Sum

```
{ int i, j, row, col;  
int A[10][10], B[10][10], C[10][10];
```

```
printf("Choose your row and column-");
```

```
scanf("%d %d", &row, &col);
```

Scan A  
matrix

```
printf("Enter A matrix's value")
```

```
for(i=0; i<row; i++)
```

```
{
```

```
for(j=0; j<col; j++)
```

```
{ printf("
```

```
printf(" A[%d][%d] = ", i, j);
```

```
scanf("%d", &A[i][j]);
```

```
}
```

```
printf("\n");
```

```
}
```

A

scanf(" Enter B matrix's value")

for(i=0; i<nrow; i++)

{ for(j=0; j<col; j++)

{

printf("B[%d][%d] = ", i, j);

scanf("%d" & B[i][j]);

}

printf("\n");

}

printf("A = ");

for(i=0; i<nrow; i++)

{ for(j=0; j<col; j++)

{

printf(" %d ", A[i][j])

}

printf("\n");

}

print B

copy → Paste. Put B in position of A.

## A Summation

Sum

```
for (i=0; i<nrow; i++)
```

```
{
```

```
    for (j=0; j<col; j++)
```

```
{
```

$$c[i][j] = A[i][j] + B[i][j];$$

```
}
```

```
}
```

```
printf("A + B = ");
```

```
for (i=0; i<nrow; i++)
```

```
{
```

```
    for (j=0; j<col; j++)
```

```
        printf("%d ", c[i][j]);
```

```
}
```

```
}
```

## A Multiplication

```
{  
int i, j, n1, c1, n2, c2;
```

```
int A[10][10], B[10][10], C[10][10];
```

```
scanf ("%d %d", &A)
```

```
scanf ("%d %d" &n1, &c1); // A matrix
```

```
scanf ("%d %d" &n2, &c2); // B "
```

```
while (c1 != n2)
```

```
{
```

```
scanf ("%d %d" &n1, &c1);
```

```
scanf ("%d %d" &n2, &c2);
```

```
}
```

For multiplication

column of first matrix

= ROW of 2nd "

1st matrix → COLUMN

2nd " → ROW

(same  
row col)

A

taking A matrix value

```
for (i=0; i<=r1; i++)  
{
```

```
    for (j=0; j<=c1; j++)  
    {
```

```
        scanf ("%d %d" & A[i][j]);  
    }
```

```
}
```

$r_0 \& c_0$	$r_0 \& c_1$	$r_0 \& c_2$
$r_1 \& c_0$	$r_1 \& c_1$	$r_1 \& c_2$
$r_2 \& c_0$	$r_2 \& c_1$	$r_2 \& c_2$

Taking B Matrix value

```
for (i=0; i<=r2; i++)
```

```
{
```

```
    for (j=0; j<=c2; j++)
```

```
        scanf ("%d %d" & B[i][j]);
```

```
}
```

( $r_1 = 1$  &  $c_1 = 2$ )

```
}
```

```
{
```

AMultiplication:2209200T

5	5	0	0
2	2	1	1

5	0	0
1	0	0
2	1	1
2	2	1



(+ 5 ; 000 > i ; 0 = i) not.

{

(+ 0 ; 000 > 0 ; 0 = 0) not

{

$$\boxed{[i]} A = [i] \boxed{e} \rightarrow 200 \text{ not}$$

A

## Transpose

(A)

0	1
1	2
2	3

Transpose

0	1	2
1	2	3
2	3	4

for(i=0; i<row; i++)

{

    for(j=0; j<col; j++)

        transpose[j][i] = A[i][j]

}

## Pointer

~~পয়েন্টা~~

computer variable এর জন্য memory address

সোজি করতে।

অর্থ যেমনই Address এর value store করে।

int x = 5; Output

printf ("%d", x); 5

int x = 5;

Output

variable name = x

value = 5

printf ("%d", &x); 100

suppose Address = 100

\* Address print করতে এল এ দিতে হব

→ Pointers এলা অনেক একটি variable এর store করে।

point এরের আগে একটি variable এর Address এর।

→ Pointers এর এরের memory address নিয়ে,

→ " অনেক একটি variable এর Address এর point এর।

P

## ✓ Declaration

datatype \*variable\_name ;

int \*p; char \*p;

(Pointer ~~to~~ variable ~~to~~)  
Address ~~of~~ ~~the~~ ~~variable~~)

## important symbols

→ & → is used to **get** **print** Address of **variable**

→ \* → Pointer or variable, or point करते

~~to value~~ ~~not store~~ ~~point~~

~~to value~~ ~~not store~~,  
~~to value~~ धूज तर नहीं मारा!

✓ & → Address

\* → Address ~~to value~~

`int x = 5; // suppose Address of x = 100`

int  $r = 10$ ; //  $r = 105$

int \*P; // pointer variable P, x, or point यहाँ

$P = \& X$ ;  $\Downarrow P$  টে  $X$  গুরুত্বে Address store করেছি।

```
printf ("%d", x); // Print→5
```

```
printf ("%d", &x); // print → 100
```

```
printf ("%d", p); // pgo value print 201  
print → 100
```

~~printf ("%d", \*p); // print content of p~~

printf ("%d", \*p); // print → 5  
P রাখে - point করছে তাৰ Address  
এতো value print হৈল।

printf ("%d", &p); // print → pointer of Addr.

P

\* Pointers হলি একটি memory management

বো = 4

গুরুত্বপূর্ণ কাজ করে, এবং যাত্রা।

\* Pointers use এখানে software fast করে করে।

printf ("%d", &x) // Address print

printf ("%d", &x) // (better)

printf ("%x", &x) // (Hexadecimal)

printf ("%x", &x) // (Hex)

// : (q \* , " b \n " ) string

← string

// : (q \* , " b \n " ) string

q ← string

// : (q \* , " b \n " ) string

## sum using pointers

```

{           ; qmst = 01 = 5, &5 = 5, tri
    int x = 5, y = 10, sum; ; 5 * , 10 * , tri
    int *P1, *P2; ; 5 = 5, x = 5
    P1 = &x; ; 5 * = qmst
    P2 = &y; ; 10 * = 10
    sum = *P1 + *P2; ; 5 + 10 = 15
    printf ("Sum = %.d", sum); → [15]
}

```

↓  
 thermal insulation would  
 & Sum); → sum variable or  
 Address.

; i ∈ {02, 04, 08, 0E, 01} = [2]A + tri  
 ; 9 \* tri

लिखिए जो संख्या नम्बर 15 है। ; [0]A \$ = 9

(int i; i > i; i = i) जो  
 ; (9 \* 1) (प्रिंट कर "5.5") #tracing  
 , जो कि ताकि यह तो करेगा // i + 9 {

P

## Swap using pointer

```

int x = 5, y = 10, temp;
int *p1, *p2; //  $p_1 = \&x, p_2 = \&y$ 
p1 = &x, p2 = &y;
temp = *p1;
*p1 = *p2;
*p2 = temp;
printf("%d %d", x, y);
}
  
```

### Access Array Pointer Element

int A[5] = {10, 20, 30, 40, 50};  


int \*p;

p = &A[0]; //  $\&A[0]$  is the address of first element

for (i=0; i<5; i++)

printf("%d", \*p);

p++; //  $p$  is incremented by 1

}

## String starting

① Copy a string using `strcpy()` function

\* `int main()`

{

    char ch;

    ch = 'A';

    ;

}

} we can store 1 character  
in 'char' data type.  
to store more characters  
we use string  
String = char dot type Array

\* {

    char ch[5];

    ch[0] = 'A';

    ch[1] = 'P';

    ch[2] = 'P';

    ch[3] = 'L';

    ch[4] = 'E';

    printf("%s", ch); // APPLE<sup>#</sup>(output)  
                        ↳ garbage value

S

କିନ୍ତୁ string କେବେ ବୁଝାଇବାରେ ଅନ୍ତର୍ଭାବରେ, last ଏବେ index ଜାରି କରିବାକୁ ପାଇଁ null character ଦିଇବାରେ ଆବଶ୍ୟକ ହେଲା।

`ch[6] = '0';`      `= '0';`

{

char ch[6];

ch[0] = 'A';

ch[1] = 'P';

ch[2] = 'P';

ch[3] = 'L';

ch[4] = 'F';

`ch[5] = '0';`

### String Functions

- `strlen()` → length (need to declare)
- `strcpy()` → copy
- `strrev()` → reverse
- `strcmp()` → compare
- `strupr()` → uppercase
- `strlwr()` → lowercase
- `strcat()` → summation

`printf("%s\n", ch);` // APPLE (Output)

ଯାହାର କିମ୍ବା କିମ୍ବା କିମ୍ବା  
କିମ୍ବା କିମ୍ବା କିମ୍ବା

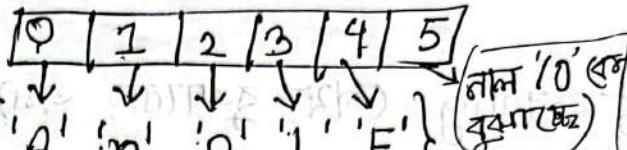
କିମ୍ବା କିମ୍ବା କିମ୍ବା

`printf("length is %d",`

### initialize

char ch[5] = {'A', 'P', 'P', 'L', 'E'}

printf("%s", ch); //APPLE



212  
char ch[] → (allocate)

{ (0) → (allocate)

char ch[] = "APPLE FRUIT"; h = [0]HS

printf("%s", ch);

} → (free)

return;

⇒ strlen() function:

{

char ch[] = "Banana Apple";

int length; // : (1)

length = strlen(ch);

printf("Length = %.d \n", length);

}

strlen(), 'ch' string का length

length variable

ये कहते हैं length variable

\* Banana

```

    ↓
B   "fint" → { create ch[] = "BANANA";
A   A           int i = 0;
N   N           while (ch[i] != '\0')
A   A           {
N   N             printf("%c", ch[i]);
A   A             i++;
}
  
```

(Fin "W2") = prints "Banana"

\* String length without strlen();

```

{ prints & bbs at Obscure
int i=0, count=0; "BANANA" = [B E]
char ch[] = "Banana"; "E" ends
for(i=0; i<=length; i++) { (L E) becomes
while (ch[i] != '\0') { (L E) becomes
for(i=0; ch[i] != '\0'; i++) { (L E) becomes
count++; } (L E) becomes
printf("length is %d", count); } (L E)
  
```

## ~~3~~ COPY STRING USING strcpy() FUNCTION

```
{  
    char ch1[] = "BANANA";  
    char ch2[30];  
    strcpy(ch2, ch1);  
    cout << "String 2 = " << ch2;  
}
```

~~strcpy(ch2, ch1);~~ मेराने देखो।  
याकृति देखो।

```
printf("String 1 = %s\n", ch1);
```

```
printf("String 2 = %s", ch2);
```

}

\* ~~concat~~ strcat(), to add 2 string

```
char ch1[] = "BANANA";  
char ch2[] = "APPLE";
```

strcat(ch1, ch2); ch2 का स्ट्रिंग (0-i) की तरफ  
ch1 का हिस्सा छोड़ देता है।

```
printf("%s", ch1);  
} // ch1 का मान ch1, प्रिंट  
// होता है। SO ch1 का print  
// होता है।
```

→ (fruits, "but it's different") printing

67

\* concat/Add 2 strings without strcat() function.

12. *Antiochus* <sup>11</sup> *Epistles* <sup>12</sup> *versus* <sup>13</sup>

3946 [ ] 18. 26913

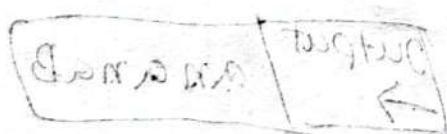
1900544 = Q 手稿

$(N_1, H_1)$  မြန်မာစီမံချက်မှုပါန်

(<sup>n</sup> Aug 3) Hohne

• ("long tail") Aborig.

primit2 92799951 ← ()



Answers - [ ] H9 name

(12) ~~versus~~

(19) "2.t") bring

\* compare 2 string strcmp()

{

char ch[] = "Banana";

char Bh[] = "APPLE";

~~int~~ int D = strcmp;

strcmp(ch, Bh);

if (D == 0)

printf("Equal");

Bh or ch അംഗീകാരം  
compare ചെയ്യുന്നതാണ്  
'0' എല്ലാ മറ്റൊരു 0 കും D കും  
ബന്ധിച്ചു

else

printf("Not Equal");

}

\* strrev() → reverse string:

{

char ch[] = "Banana";

Output  
→ ananab

strrev(ch);

printf("%s", ch);

}

## reverse without strrev()

"BANANA" = [ ] [ ] [ ] [ ] [ ] [ ] [ ]

"EARTH" = [ ] [ ] [ ] [ ] [ ] [ ] [ ]

[ ] [ ] [ ] [ ] [ ] [ ] [ ]

The great and the earth ; (the great) opposite

the great and the earth ; (the great) opposite

The great and the earth ; (the great) opposite

The great and the earth ; (the great) opposite

Checking a string is palindrome or not,

main string or reverse string or ~~main~~ string main string or never fluctuate else or palindrome.

char ch1[] = "Banana";

~~char ch1[];~~

char ch2[30];

int compare;

strcmp(ch2, ch1);

strncpy(ch2);

compare = strcmp(ch1, ch2);

if (compare == 0)

printf("Palindrome")

else

printf("Not palindrome")

return 0;

}

## \* swap string

```
char ch1[] = "Banana";
```

```
char ch2[] = "APPLE";
```

```
char temp[30];
```

```
strcpy(temp, ch1); // ch1 to temp
```

~~swap(ch1, ch2)~~

```
strcpy(ch1, ch2); // ch1 & ch2 to temp
```

```
strcpy(ch2, temp); // ch2 to temp/ch1 to temp
```

```
printf("%s %s", ch1, ch2)
```

}

After Swap

Ans ch1 = APPLE

ch2 = Banana

\* strupr(), strlwr(),

char ch[] = " BANana";

strupr(ch);  
printf("%s", ch); → BANANA

strlwr(ch);  
printf("%s", ch); → banana

} ++ timecomplexity

\* vowel, consonent, word, number:

char ch[100], c;

~~scanf~~ int i, consonent, vowel, word, digit, other;

~~gets(str)~~

gets(ch); // scanf("%s", str);

i = consonent = vowel = word = digit = other = 0;

while (ch[i] != '\0') // ৰ তক্ষণ string এর last character  
// মাঝে গতক্ষণ ফলো।

{  
; (lower("b") == lower("b")) ? true,

~~if~~ {  
    if ( $c == 'c' \& c <= 'z'$ ) ++vowel;  
    else if ( $c >= 'A' \& c <= 'Z'$ ) ++consonant;  
    else if ( $c >= '0' \& c <= '9'$ ) ++digit;  
    else if ( $c == '-'$ ) ++other;

if ( $c == 'a' \& c <= 'e'$ ) ++vowel;  
    else if ( $c == 'A' \& c <= 'E'$ ) ++consonant;  
    else if ( $c == 'i' \& c <= 'o'$ ) ++vowel;  
    else if ( $c == 'U' \& c <= 'O'$ ) ++consonant;

else if ( $(c >= 'a' \& c <= 'z') \& (c >= 'A' \& c <= 'Z')$ )  
    consonant++;

else if ( $c >= '0' \& c <= '9'$ )  
    number++;

digit++;

else if ( $c >= 'a' \& c <= 'z'$ )

else if ( $c == 'I'$ )

word += c;

else if ( $c == '0' \& c <= '9'$ )  
    word += c;

printf (" vowel = %d ", vowel);

~~S~~ printf (" consonent = \"%d\" consonent);

----- digit -----, digit );

----- word -----, (word+1));

----- Other -----, other );

}

to Digit, capital, small :

```
{  
char ch[100], c;  
int i=0, cap=0, sm=0, dig=0;  
gets(ch);  
while (ch[i]!='\0')  
{  
    c = ch[i];  
    if (c>=65 && c<=90)  
        cap++;  
    else if (c>=97 && c<=122)  
        sm++;  
    else if (c>=48 && c<=57)  
        dig++;  
}
```

printf (" Capital = \"%d\", cap);  
----- Small -----, sm));  
----- Digit -----, dg));

}

End String

# Function Starting

Sub :

Day

Time:

Date: / /

#include <stdio.h>

int sum (int a, int b);  
{  
 return (a+b);  
}

user define function

main  
function

\* program starting main function  
(main का शुरू होता है)  
  
int main ()  
{  
 int num1, num2;  
 scanf ("%d %d", &num1, &num2);  
 sum (num1, num2);  
 printf (" Sum is %.d ", sum (num1, num2));  
}  
sum (10, 20);  
sum (5, 10);  
sum (20, 10);  
sum (10.5);

# Function

Sub:

Day

Time:

Date: / /

② void PrintHello() → Statement term  
↓ ↓  
return type      function name  
→ 2 Parentheses (param)

1 Full Process of function Declaration

or " prototype"

② void PrintHello  
void printHello () {  
    printf(" Hi \n");  
}

2 This process name function definition

Given off base of base ← Library  
Given off function " " ← system

Sub:

Actual Parameter

Formal Parameter

Day

No.

Date: / /

Time:

or

Arguments

or

Parameters

or

Local Variable

Definition

Argument  
Parameter

int main ()

{ Local Variable Declaration

int Number; ← Local Variable declaration

calling the "call" function

call (Number); ← Argument

return 0;

}

} ( ) calling local variable

void printHello ( )

\* void call (int p)

Parameters

{

}

Argument → used to send the value

Parameter → " " received the value

## Structure & function

Sub :

Day

--	--	--	--	--	--	--	--

Time :

Date : / /

\* void Semi (struct student s1)  
data type variable

Struct student {

int roll;

int age;

}

void PrintStructure (struct student s1)

{

printf (" Roll = %d ", s1.roll);

printf (" Age = %d ", s1.age);

}

int main () {

struct student s1 ;

s1.roll = 1236;

s1.Age = 21;

call by value

PrintStructure (s1);

}

## Sub: Comparison

Day: / /  
Time: / / Date: / /

same code

~~#include <stdio.h>~~

{ (main)

    |

    |

    |

int main()

{ (main)      ~~Struct Person~~      ~~variables~~      ~~variables~~

    Struct Person P<sub>1</sub> = {25, 2500};

    P<sub>1</sub>.age = 25;      P<sub>1</sub>.Salary = 2500;

    Struct Person P<sub>2</sub> = {15, 1500};

    If (P<sub>1</sub>.age == P<sub>2</sub>.age & P<sub>1</sub>.Salary == P<sub>2</sub>.Salary)

        printf ("Both Equal");

    else

        printf ("Not Equal");

}

{100, 10} = First record

{200, 20} = Second record

# Structure

Sub:

Day

Time:

Date: / /

W

Q

struct student

{

int roll;

float egpa;

char name[50];

}

int main()

{

struct student

data type

Structure

attribute found

s<sub>1</sub> ;  
variable

Declare.

strcpy(s<sub>1</sub>.name, "Sami Sami"); value assign

s<sub>1</sub>.roll = 10;

(f+i : i>i ; o=i) not

s<sub>1</sub>.egpa = 4.00;

Student { }  
(f+i)(#) attribute

printf("%s", s<sub>1</sub>.name);

-- ("%.f", s<sub>1</sub>.egpa);

: (98.00 : 172 --> 100)

}

~~OR~~ int main()

1

Struct Student  
D.T.

$$\underline{s}_1 = \left\{ 10, 3.86, "Semi" \right\}$$

```
cout << s1.name << endl;
```

~~200-1050~~ S1: 17011

— — — S<sub>1</sub> · egpa — — — Probuta hantae

~~scanf("%[^m]s", string);~~

```
int main()
```

street student S[2];

```
for(i=0; i<2; i++)
```

{ // student ~~(i+1)~~ values

```
fflush(stdin); //scanf("\n");
```

flash (flashing) *flashing*

*if you do "what you can"*

```
gets(s[i].name); // fgets(s[i].name
```

June 22, 1992 - 1992

scanf( " %d ", &s[i] );

seen = - - - s[i].egpa);

1

Print

```
for (i=0; i<2; i++)
```

{

```
    Print ("Student %d Name=%s", i+1);
```

```
    puts (s[i].name);
```

~~Print ("Student %d roll, i+1);~~~~Print ("Student %d roll = %d", i+1, s[i].roll)~~~~Print ("Student %d cgpa = %.2f", i+1, s[i].cgpa)~~

(  
    )  
    {

\* struct student {

char name[100];

};

// main function

\* s1.name = "Sami Sami"

✓ strcpy (s1.name, "Sami Sami");

sean

☛ ~~scanf(" %[^\\n]s", s1.name);~~ ✓

✓ gets(s1.name); ✓

☛ fgets(s1.name, 100, ~~stdin~~ {stdin}); ✓

1 एक string input करें

fflush(stdin);

gets(s1.name);

scanf("%")

gets(stdin);

scanf("\n");

gets(stdin);

## File এর সtructure.

বিধাতে data store রয়েছে।

## FILE \*name\_of\_file;

### Functions:

To write something

- #include <stdio.h>
- fputc()
- fputw()
- fputs()
- fprintf()
- fprintf()

to read something

- fgetc()
- fgetw()
- fgets()
- fscanf()
- fread()

### file open / তৈরি করতে fopen()

fopen("file name", "file mood");

a = add - করা (a=append)

#include < stdio.h >

int main ()

{

FILE \*file ;

fopen(" ~~file~~ text .

file=fopen(" test.txt ", "w");

if (file == NULL)

printf(" File not exist ");

else {

printf (" File is created );

fclose(file); // file close করো

}

r = read করতে

w = write

যদি কোন File create

না থাকে তবে w use

কোর্ট আসলে create

হবে নিচে

যদি File create হয়

তবে file এ লাই character

থাকবে।

Sub: \_\_\_\_\_

`fputc(); (character, filename);`  
এই মিত্র করে এবং একটি character write কর

রাখ ।

\* `w = write করো (নতুন বা write করো)`

`r = read`

`a = append (চেস্ট করে আপডেট করো)`

\* ~~string~~ `w`

\* file এ string write করতে `fputs();`

`(string name, filename);`

or `("string", filename);`

{

Sub : \_\_\_\_\_

Day

--	--	--	--	--	--	--

Time : \_\_\_\_\_

Date : / /

# Pointer Arithmetic

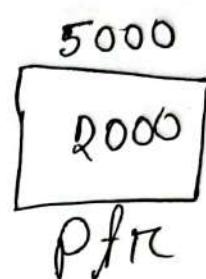
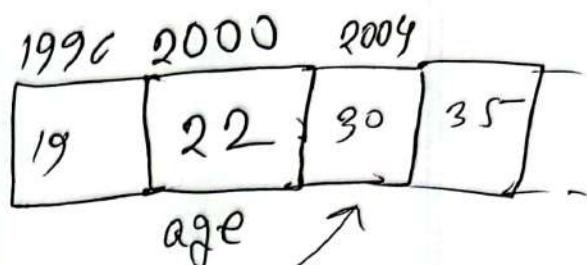
Sub:

Day: \_\_\_\_\_  
Time: \_\_\_\_\_ Date: / /

int age = 20;

int \*pfrc = ~~&~~ ~~&~~ age;

pfrc++;



~~we can~~ printf("%d", \*pfrc);



we can <sup>difference</sup> compare 2 pointers,

int A = 20; B = 10;

int \*P = &A;

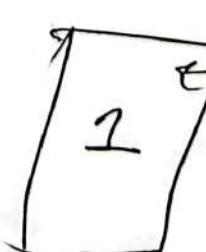
int \*Q = &B;

printf("%d", P-Q)

P = 20004

Q = 2000

$P-Q = 4$   
1 integer



### Q) comperce

int \*p = &A

int  
 A = 20  
 B = 10

( ) main fun

int \*q = &B

Print("U", p = q); → 1 - true  
 P = q

if (i > i : 0 = i) if  
 if ((it == 0) , b.v ) pass;

### Q) Pointer + Array :

int arr[] = { };

int \*ptr = &arr[0];

or int \*ptr = arr;

Same

Sub:

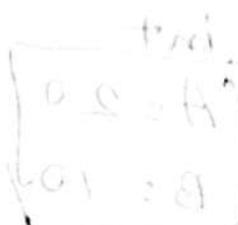
# Scan & Print in array using pointer taking arr

Time:

Date: / /

# -----

int main()



arr[0] = 0 1 2 3 4 5

arr[1] = 0 1 2 3 4 5

{ int arr[5]; }

arr[2] = 0 1 2 3 4 5

int \*ptr = &amp;arr; [arr[0] is Add]

for (int i=0; i&lt;5; i++) { printf("%d", \*(ptr+i)); }

for (int i=0; i&lt;5; i++)

ptr = (\*arr[i]) \* (ptr+i);

arr[0] = 0 1 2 3 4 5

arr[1] = 1 2 3 4 5

# Pointers + Funktion + Arrays

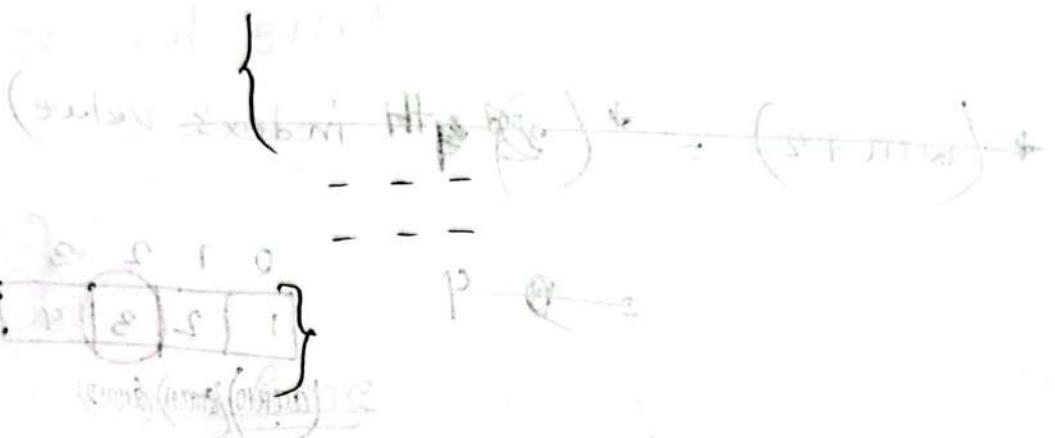
Sub:

Day

Time:

Date: / /

# void sami (int \* arr, int sz) {



int main ()

{

$(\text{arr} \text{ at } e)^* = (\text{arr})^*$

call by reference

sami (arr, n);

$\&\text{arr}[0]$  Num of Array elements

}

$\text{arr} = a * \text{hi} \quad // [\text{C}] \text{arr} & = a * \text{hi}$

$$e = (e + n) *$$

$$\text{arr} = (e + n) *$$

Sub : \_\_\_\_\_

Day \_\_\_\_\_

Time : / /

Date : / /

Q)

 $\text{int } *(\text{arr}[]) = \{1, 2, 3, 4, 5\}$ 

1	2	3	4	5
0	1	2	3	4

~~$*(\text{arr}[2]) = *(\text{arr}[\text{index} + 1])$  index's value~~



0	1	2	3	4
1	2	3	4	5

(arr[0]) (arr[1]) (arr[2]) ...

⇒  $*(\text{arr}[2]) = *(\text{2}^{\text{nd}} \text{ index})$

$$= 3$$

⇒  $*(\text{arr}[5]) = *(\text{5}^{\text{th}} \text{ index})$

= Garbage value. ~~error~~

Q)

 $\text{int } *(\text{arr}[]) = \{1, 2, 3, 4, 5\}$ 

$\text{int } *a = &\text{arr}[0] // \text{int } *a = \text{arr};$

$$*(*a + 2) = 3$$

$*(*a + 5) = \text{Garbage.}$

# Structure

Sub :

Day

--	--	--	--	--	--

Time :

Date : / /

## \* Pointers + Structure

struct student {

    - - - -  
    - - - -

};

### // main func

main() {  
    // code  
    {  
        struct student s1; // local variable

        struct student s1;  
        D.T

        struct student \*p;

        struct student s1;  
        D.T

        struct student \*p;

        p = & s1; // address of s1

    } // main function

    ("Name : (Rishabh)", "a = small") // printing

Sub : \_\_\_\_\_

Day \_\_\_\_\_

Time : / /

Date : / /

```
struct student {  
    char name[50];  
    int roll;  
    int age;  
};
```

// main

```
{  
    struct Student s1; // D.T Var  
    strcpy(s1.name, "Samir"); // value assign  
    s1.roll = 10; // attribute  
    s1.age = 20; // attribute
```

```
struct student *ptr; // D.T Pointer
```

```
ptr = &s1; // s1 is Address // point variable
```

Print

```
printf("Name = %s", (*ptr).name);
```

```
----- ("Roll = %d", (*ptr).roll);
```

Sub : \_\_\_\_\_

Day

--	--	--	--	--	--	--

Time :      Date : / /

~~OR~~

~~printf("Roll = %.d", ptr->roll);~~ ~~(\*ptr).roll~~  $\Rightarrow$  ~~ptr->roll~~

printf("Name=%s", ptr->name);