

* Why we use C?

⇒ 'C' has been used successfully for every type of programming problem and ~~has~~ thin renown.

- ① The portability of the compiler.
- ② The standard library concept.
- ③ A powerful and varied repertoire of operators.
- ④ An elegant syntax.
- ⑤ ready access to the hardware when needed.
- ⑥ The ease with which application can be optimized by hand-coding isolated procedures.

* Application of 'C'.

⇒ Some examples of the use of 'C' might be.

- ① operating system
- ② Language compilers
- ③ Assemblers
- ④ Text Editors
- ⑤ print Spoolers
- ⑥ Network Drivers
- ⑦ Modern program
- ⑧ Data Bases
- ⑨ Language Interpreters
- ⑩ Utilities.

1. Input Output

① Write a program that prints your name :-

→ `#include<stdio.h>`

`int main()`

{
`printf ("My name is Md. Belayet Honnain");`
`return 0;`

}

["printf" ~~and~~ library
function]

Comment

① Single Comment

~~(to)~~ ~~(at)~~ // ~~f625~~
f1C 2C

② multiple Comment

~~(to)~~ ~~(at)~~ /* ...
*/ f625 f1C 2C

Ans: // Write a program that prints my program

`#include<stdio.h>`

`int main()`

{
`printf ("My name is Md. Belayet Honnain");`
`return 0;`

}

Escape Sequence

→ C program ରେ ମିଳିବାଲା Backslash Character ରୂପରେ କଣ ହୁଏ ଯାଏ ଆର୍ଟିଫିଶିଆଲ ରୂପରେ ଲିଖାଇଲେ କାହା କାହା !

→ ଅଣିଗତ Backslash (\) ରୁଥ୍ୟାରୁ ମିଳି ଏବେଳି କ୍ଷାନ୍ତୋଫ୍ ରୂପରେ କଣ ହୁଏ !

Character Escape Sequence	Meaning	Character Escape Sequence	Meaning
\a	(Alert) Bell	\v	Vertical Tab
\b	Back Space	\0	Null Character
\f	Form Feed	\'	Single Quote
\n	New Line	\"	Double Quote
\r	Carriage Return	\N	Backslash
\t	Horizontal Tab	\?	Question Mark

ଉଦାହରଣ : #include<stdio.h>

int main()

{ printf(" My name is Belayet Honnain \n");

printf(" My father name is Nurul Islam \n");

printf(" My mother name is Aleya Begum \n");

return 0;

3

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

40912

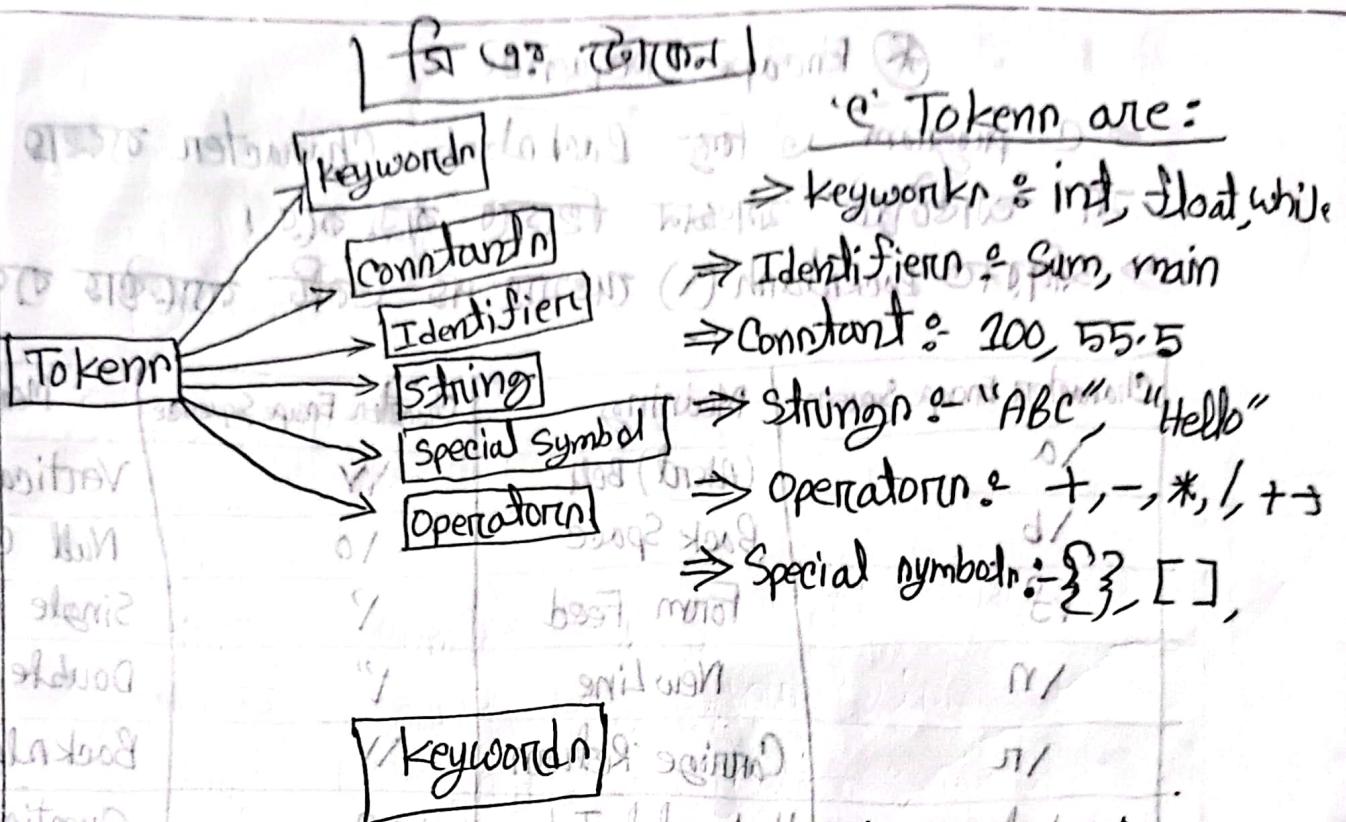
40912

40912

40912

40912

40912



⇒ There are 32 keywords in 'C' programming.

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

Data Type

* Variable declaration syntax:

Data type	example	keyword	Storage size	Format Specifier
character data	'a', 'n', '1'	char	1 byte	%c
whole numbers	125, 200	int	4 bytes	%d
Floating point numbers	6 decimal places	float	4 bytes	%f
Double numbers	15 decimal places	double	8 bytes	%lf

1.2 An integer variable n contains 5. Write a program that print the value of n.

→ /* An integer variable n contains 5, write a program that print the value of n

```
/*
#include <stdio.h>
int main()
{
    int n=5;
    printf("Number is %d",n);
    return 0;
}
```

9987 090

1.2.2 Write a program that prints a floating double number and character,

⇒ #include <stdio.h>

int main()

{ float num1 = 10.5;

double num2 = 10.52555542;

char ch = 'a';

printf ("num1 = %f", num1);

printf ("num2 = %lf", num2);

printf ("%c", ch);

return 0;

}

(1.3)

1.3 write a program that reads an integer number.

⇒ #include <stdio.h>

int main()

{ int num;

printf ("Please enter an integer = ");

scanf ("%d", &num);

printf ("You have pressed = %d", num);

return 0;

}

① 1.5 Write a program that read and display a floating point number.

⇒ #include <stdio.h>
int main()

{ float num;
printf ("Please enter a floating number = ");
scanf ("%f", &num);

printf ("You pressed = %f", num);
return 0;

① 1.6 Write a program that read and display long number.

⇒ #include <stdio.h>
int main()

{ float num;

printf ("Please enter a long number = ");

scanf ("%

{ long long-num

printf ("Please enter a long number = "));

scanf ("%lld", &long-num);

printf ("You have pressed = %lld ", long-num);

return 0;

3

1.7 Write a program that find the size of int, float, double and char data type.

→ #include <stdio.h>

Input: int main()

```
{  
    int i;  
    float f;  
    double d;  
    char c;  
    printf ("Size of int = %d bytes\n", sizeof(i));  
    printf ("Size of float = %d bytes\n", sizeof(f));  
    printf ("Size of double = %d bytes\n", sizeof(d));  
    printf ("Size of char = %d bytes", sizeof(c));  
}
```

}

Output:

Size of int = 4 bytes

Size of float = 4 bytes

Size of double = 8 bytes

Size of char = 1 byte

{

1.8 Write a program that read ASCII value and display equivalent character.

⇒ Code:

```
#include<ntdio.h>
int main()
```

```
{ int n;
printf("Enter any ASCII value: ");
scanf("%d", &n);
printf("The ASCII character is: %c", n);
```

}

Output: Enter any ASCII value : - 97

The ASCII character is : a

1.9 Write a program that read ASCII character and display equivalent ASCII value.

⇒ Code:

```
#include<ntdio.h>
```

```
int main()
```

```
{ printf("Enter
```

char ch;

printf("Enter any ASCII character: ");

scanf("%c", &ch);

printf("The ASCII value is: %d", ch);

}

Output:

⇒ Enter any ASCII character

: A

The ASCII value is : 65

: 65

: 65

: 65

: 65

: 65

: 65

: 65

: 65

: 65

: 65

: 65

: 65

1.11 Write a program that read any lower case character and display in upper case without library function.

⇒ Code:

```
#include <ntdlo.h>
int main()
```

{

char lower;

printf ("Enter any lower case letter : ");

scanf ("%c", &lower);

printf ("The upper case letter is : %c", lower - 32);

}

Output: Enter any lower case letter : - a

The upper case letter is : A

lower case ascii value 97
upper case ascii value 65
~~32~~ 65 + 32 = 97

1.12 Write a program that read any upper case letter and display in lower case without library function.

⇒ Code:

```
#include <ntdlo.h>
int main()
```

{

char upper;

printf ("Enter any upper case letter : ");

scanf ("%c", &upper);

printf ("The lower case letter is : %c", upper + 32);

3

⇒ Output:

Enter any upper case letter : - E

The lower case letter is : e.

1.11(a) Write a program that read any lower case letter and display in uppercase with library function.

⇒ Code: #include <stdio.h>

int main()

{

char lower, upper;

printf("Enter any lowercase letter: ");

scanf("%c", &lower);

upper = toupper(lower);

printf("Uppercase letter = %c", upper);

3

Output: Enter any lowercase letter: - a

Uppercase letter = A

1.12(a) Write a program that read any uppercase letter and display in lowercase with library function.

⇒ Code: #include <stdio.h>

int main()

{

char lower, upper;

printf("Enter any uppercase letters: ");

scanf("%c", &upper);

lower = tolower(upper);

printf("Lowercase letter = %c", lower);

3

Output: Enter any uppercase letter: - A

lowercase letter = A

④ Some formate specifier and Example

Code	Description	Example	Result
%d, %i	print any integer	printf "%d" 5	5
%x, %o	print any integer in Hex or Octal format	printf "%x" 255	ff
%s :	print any string	printf "%s" "ABC"	ABC
%f	print any float-point number	printf "%f" 2.1M	1.100000
%c	print any character	printf "%c" '\097'	a
%b	print any Boolean	printf "%b" false	false
%o	print any object	printf "%o" (1,2)	(1,2)
%A	print anything	printf "%A" (1,2)	(1,2)

1.13 Write a program that read any decimal number and display equivalent octal number.

Code: #include<stdio.h>

```

int main()
{
    int number;
    printf ("Enter any decimal number : ");
    scanf ("%d", &number);
    printf ("Octal number = %o", number);
}
  
```

Output: Enter any decimal number : 25

Octal number = 31.

A = notok sambewah

1.14 Write a program that read any decimal number and display equivalent Hexadecimal number.

⇒ Code:

```
#include<ntdio.h>
int main()
{
    int number;
    printf("Enter any decimal number: ");
    scanf("%d", &number);
    printf("Hexadecimal number = %x", number);
}
```

Output: Enter any decimal number = -55

Hexadecimal number = 3F

1.15 Write a program that read any octal number and display equivalent decimal number.

⇒ Code:

```
#include<ntdio.h>
int main()
{
    int number;
    printf("Enter any octal number: ");
    scanf("%o", &number);
    printf("Decimal number = %d", number);
}
```

Output: Enter any octal number : - 63

Decimal number = 51

SPI = number conversion

1.17) Write a program that read any Hexadecimal number and display equivalent decimal number.

⇒ Code:

```
#include <stdio.h>
int main ()
{
    int number;
    printf ("Enter any Hexadecimal number : ");
    scanf ("%x", &number);
    printf ("Decimal number = %d", number);
}
```

Output: Enter any Hexadecimal number : - a

Decimal number = 10

1.17.a) Write a program that read any octal number and display equivalent Hexadecimal number.

⇒ Code:

```
#include <stdio.h>
int main ()
{
    int number;
    printf ("Enter any octal number : ");
    scanf ("%o", &number);
    printf ("Hexadecimal number = %x", number);
}
```

Output: Enter any octal number : 502

Hexadecimal number = 142

1.17(b) Write a program that read any Hexadecimal number and display equivalent octal number.

⇒ Code :-

```
#include <stdio.h>
int main()
{ int number;
```

```
printf (" Enter any Hexadecimal number : ");
```

```
scanf ("%x", &number);
```

```
printf (" Octal number = %o", number);
```

Output: Enter any hexadecimal number: F25

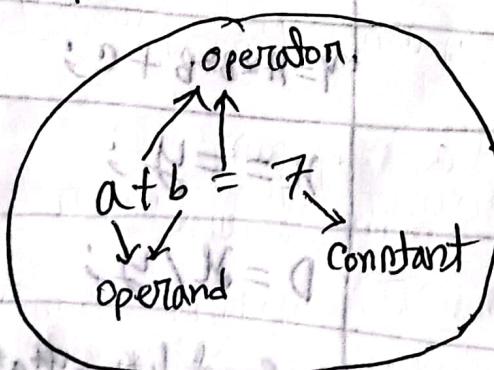
Octal number = 745.

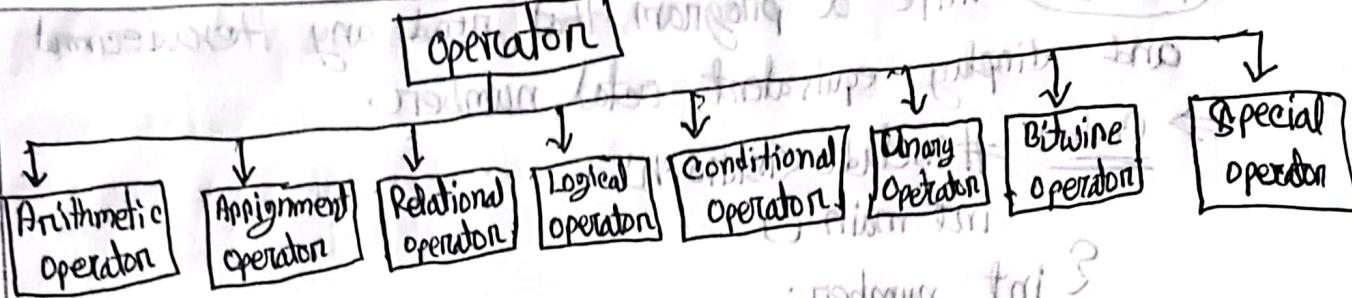
Operator, Operand and Expression

① Operator कि ?

⇒ ये असाधा गणितीय व सीखिए लाभ नियमों द्वारा अनुरूप गणितीय विकल्प हैं (जैसे : +, -, *, /) जहाँ पर ये अपेक्षित

~~Specified operator~~ होते हैं।





Arithmetic Operator

Operator	Work	Example	Result
+	Addition	$x = 15 + 6$	$x = 21$
-	Subtract	$x = 15 - 6$	$x = 9$
*	Multiplication	$x = 15 * 6$	$x = 90$
/	Division	$x = 15 / 6$	$x = 2$
%	Modulus	$x = 15 \% 6$	$x = 3$

④ Mathematical expression & Equivalent C expression

Mathematical expression:	Equivalent C expression:
$x = a^2 - 2ab + b^2$	$x = a * a - 2 * a * b + b * b;$
$y = AB^2 + C$	$y = A * B * B + C;$
$x = y$	$x = y;$
$D = x / y$	$D = x / y;$
$D = \sqrt{b^2 - 4ac} / 2a$	$D = \text{Sqr} \{ b * b - 4 * a * c \} / 2 * a;$

2.2 Write a program that read two integer numbers and display their sum.

```

⇒ Code: #include <stdio.h>
int main()
{
    int num1, num2, sum;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);
    sum = num1 + num2;
    printf("The sum is: %d", sum);
}

```

Output: Enter two numbers n : 10, 19
The num is 19.

② Write a program that subtracts two integers.

```

Code: #include <ntdib.h>
int main()
{
    int num1, num2, sub;
    printf ("Enter two numbers : ");
    scanf ("%d %d", &num1, &num2);
    Sub = num1 - num2;
    printf ("The subtraction = %d", sub);
}

```

Output: Enter two numbers : - - 8, 6
The subtraction is = 2.

2.3 Write a program that read two integer and display product.

⇒ Code: #include <stdio.h>
int main()
{ int num1, num2, mul;
printf("Enter two numbers: ");
scanf("%d %d", &num1, &num2);
mul = num1 * num2;
printf("The two number product is = %d", mul);
}

Output: Enter two numbers: 40, 10
The two number product is = 400

* write a program that taken two integer and display sum and average.

⇒ Code: #include <stdio.h>
int main()
{ int num1, num2, sum;
float avg;
printf("Enter two numbers: ");
scanf("%d %d", &num1, &num2);
sum = num1 + num2;
printf("The sum is = %d\n", sum);
avg = (float)num / 2; // type casting;
printf("The average is = %.2f", avg);
}

3

Output: Enter two numbers = 7, 8
The sum is = 15
The average is = 7.5.

* Write a program that read two integer number and display their sum, sub, multiplication, division.

⇒ Code: #include <stdio.h>

```
int main()
```

```
{ int num1, num2, sum, sub, mul, dev;
```

```
float dev;
```

```
printf ("Enter two numbers : ");
```

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

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

```
sub = num1 - num2;
```

```
mul = num1 * num2;
```

```
dev = num1 / num2; // type casting
```

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

```
printf ("Sub = %d", sub);
```

```
printf ("Multiplication = %d", mul);
```

```
printf ("Division = %.2f", dev);
```

```
}
```

Output: Enter two numbers = 7, 5

Sum = 12

Sub = 2

Multiplication = 35

Division = 1.4

2.4 and 2.6 included

④ Write a program that read two numbers and display their division and remainder.

⇒ Code: #include <stdio.h>

```
int main()
{
    int num1, num2, remainder;
    float div;
    printf("Enter two numbers: ");
    scanf("%d %d", &num1, &num2);
    div = (float) num1 / num2; // type casting
    remainder = num1 % num2;
    printf("Division = %.2f\n", div);
    printf("Remainder = %d", remainder);
}
```

Output: Enter two numbers:

division = 3.29

Remainder = 2

⑤ Write a program that read and divide two floating point numbers.

⇒ Code: #include <stdio.h>

```
int main()
{
    float num1, num2, div;
    printf("Enter two floating numbers: ");
    scanf("%f %f", &num1, &num2);
    div = num1 / num2;
    printf("Division = %.2f\n", div);
```

Output: Enter two floating numbers $a = 35.4$ and $b = 12.3$

Division = 2.91 which has significant 2 digits

* Write a program that calculate the area of triangle

→ Code:

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

```
int main()
```

```
{ float base, height, area;
```

```
printf("Enter the triangle base : ");
```

```
scanf("%f", &base);
```

```
printf("Enter the triangle height : ");
```

```
scanf("%f", &height);
```

```
area = 1/2 * base * height (float) 1/2 * base * height
```

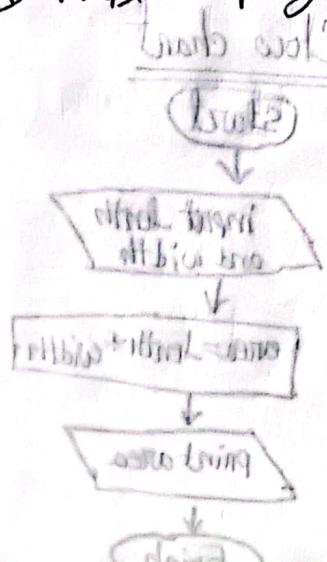
```
printf("Area of triangle = %f", area);
```

Output: Enter the triangle base: 3

Enter the triangle height: 5.6

Area of triangle = 8.4

* Write program that solution to a/b with



multifacult

trust

Abies Den brak trust bugi

Abies * Abies = 1000

Abies salt galqib

Nainit

④ Write a program that read the length of the three sides of a triangle and display its area.

⇒ Code:

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

```
int main()
```

```
{
```

```
    double a, b, c, s, area;
```

```
    printf("Enter the length of the three sides: ");
```

```
    scanf("%f %f %f", &a, &b, &c);
```

```
    s = (a+b+c)/2
```

```
    area = sqrt(s*(s-a)*(s-b)*(s-c));
```

```
    printf("The area of the triangle = %.2f\n", area);
```

3

~~Output:~~ Output: $\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$

Enter the length of the three sides : 5.7, 4.5, 3

The area of the triangle = 6.70

⑤ Write algorithm, flow chart and programming to find the area of rectangle.

⇒ Algorithm

① Start

② Input length and width

③ area = length * width

④ display the area

⑤ Finish

flow chart

start

input length
and width

area = length * width

print area

Code: #include <stdio.h>

```
int main()
{
    float length, width, area;
    printf("Enter length = ");
    scanf("%f", &length);
    printf("Enter width = ");
    scanf("%f", &width);
    area = length * width;
    printf("The area is = %.2f\n", area);
}
```

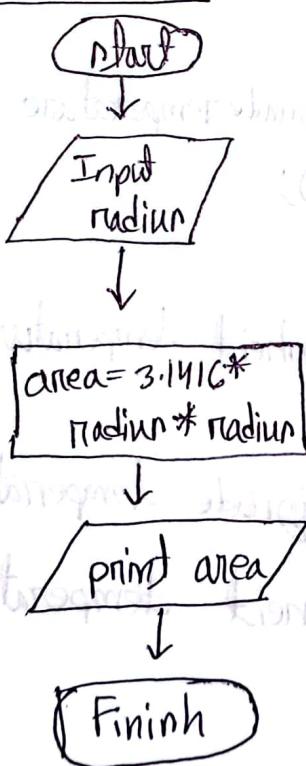
Output:
Enter length = 6
Enter width = 5
The area is = 30

* Write a ~~program~~ algorithm, flow chart and code to find the area of circle.

→ Algorithm

- ① Start
- ② Input radius r
- ③ area = πr^2
- ④ Display the area
- ⑤ Finish

Flow chart



Code:

```
#include <stdio.h>
int main()
{
    float radius, area;
    printf("Enter Radian = ");
    scanf("%f", &radius);
    area = 3.1416 * radius * radius;
    printf("The area is = %.2f", area);
}
```

Output: Enter Radian = 4.6
The area is = 66.48

5 D 1.8
no v.
to

Write algorithm, flow chart and code to express temperature in centigrade scale to Fahrenheit scale.

Algorithm:

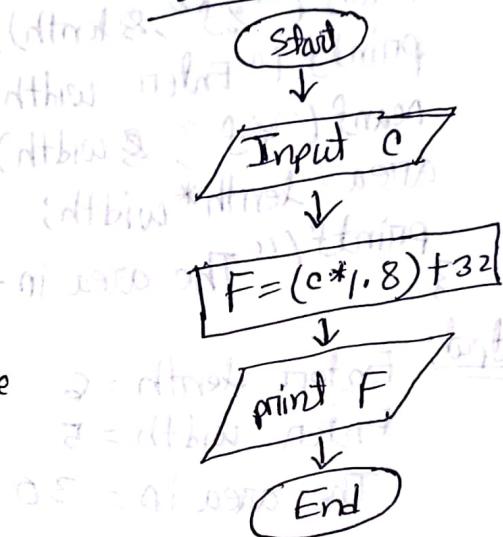
① Start
② Input centigrade temperature

③ Use formula of Fahrenheit
 $F = (C * 1.8) + 32$

④ Display the centigrade temperature

⑤ End

Flow chart



Code:

```
#include <ntdio.h>
int main()
{
    float C, F;
    printf("Enter centigrade temperature : ");
    scanf("%f", &C);
    F = (C * 1.8) + 32;
    printf("The fahrenheit temperature is = %.2f\n", F);
}
```

Output: Enter centigrade temperature = 36

The fahrenheit temperature is = 96.80

Q) Write a program to express temperature in Fahrenheit
scale to centigrade scale.

$\Rightarrow \#include <stdio.h>$
int main ()

{ float c, f;

printf ("Enter Fahrenheit temperature : ");
scanf ("%f", &f);

$$c = \frac{5}{9}(F - 32) / 1.8$$

printf ("The centigrade temperature is = %f\n", c);
3

Output: Enter Fahrenheit temperature : 53.60

The centigrade temperature is = 12.000000

We know

$$\frac{9}{5} = \frac{F-32}{C}$$

$$\text{on}, \frac{9}{5}C = F - 32$$

$$\text{on}, C = \frac{F-32}{1.8}$$

Q) How to swap two numbers with temporary variable

$\Rightarrow \#include <stdio.h>$
int main ()

{ int num1 = 10;

int num2 = 5;

int temp;

temp = num1;

num1 = num2;

num2 = temp;

printf ("num1 = %d\n", num1);

printf ("num2 = %d\n", num2);

3

Output: num1 = 5

num2 = 10

[swapping মান হকি ও স্বতু
value অন্তর্ভুক্ত করেন কো]

Q How to swap two numbers without temporary variable.

⇒ code: #include <stdio.h>
int main()

int num1 = 45

int num2 = 20

num1 = num1 + num2;

num2 = num1 + num2;

num1 = num1 - num2;

printf ("num1=%d\n", num1);

printf ("num2=%d\n", num2);

Output: num1=20 in environment string 0 NT

num2=45

U.1.3 Q Write algorithm, flow chart and program to express

$ax^2+bx+c=0$ quadratic equation.

⇒ We know, the formula of $ax^2+bx+c=0$

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

$$\therefore x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$\text{and } x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Algorithm:

① Start

② Input a, b, and c

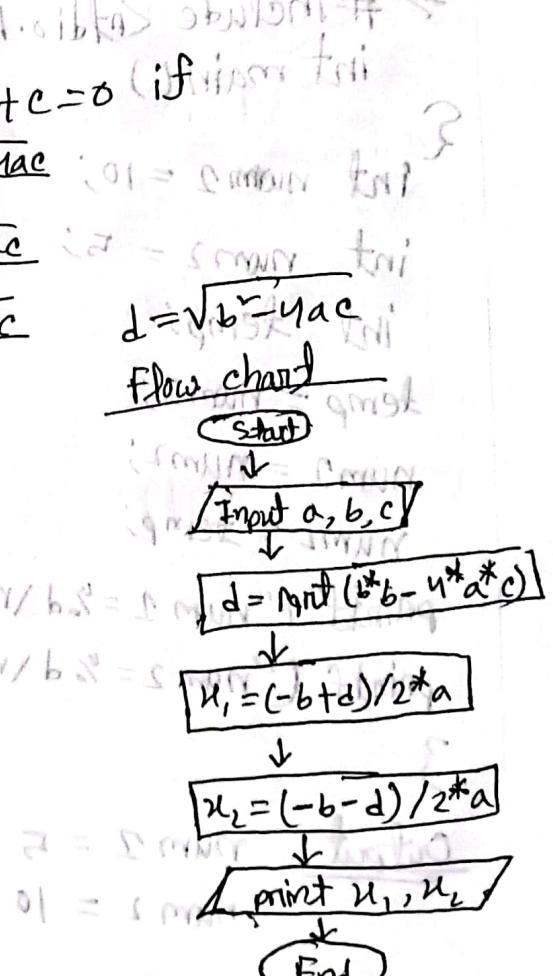
③ calculate $d = \sqrt{b^2 - 4ac}$

④ calculate $x_1 = (-b + d) / 2a$

⑤ calculate $x_2 = (-b - d) / 2a$

⑥ print x_1, x_2

⑦ End



Code: #include <stdio.h>

```
int main()
{
    double a, b, c, d, x1, x2;
    printf("Enter a, b, and c = ");
    scanf("%lf %lf %lf", &a, &b, &c);
    d = a*a - b*b + 4*a*c;
    x1 = (-b + d) / 2*a;
    x2 = (-b - d) / 2*a;
    printf("x1 = %lf\n", x1);
    printf("x2 = %lf\n", x2);
}
```

Output: Enter a, b, and c = 2 5 2

$$\begin{aligned}x_1 &= -2 \\x_2 &= -8\end{aligned}$$

Math.h

④ Absolute value

Write a program that read any integer number and display absolute value.

3.1

Code: #include <stdio.h>

```
int main()
```

```
{
    int result = abs(-7);
    num;
    printf("Enter any number : ");
    scanf("%d", &num);
    result = abs(num);
    printf("Absolute number is = %d\n", result);
```

Output:

Enter any number = -56

Absolute number is = 56.

3.13 Write a program that read any number and display its square root.

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

```
int main()
```

```
{
```

```
double num;
```

```
printf("Enter any number :");
```

```
scanf("%lf", &num);
```

```
num = sqrt(num);
```

```
printf("The square root value is = %lf\n", num);
```

3

Output: Enter any number: 54

The square root value is = 7.348469

3.14 Write a program that read two numbers (x, y) and display the value of x^y .

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

```
int main()
```

```
{
```

```
int x, y;
```

```
printf("Enter x and y :");
```

```
scanf("%d %d", &x, &y);
```

```
double result = pow(x, y);
```

```
printf("%d^%d value is = %.2f\n", result);
```

3

Output: Enter x and y : 5 2

x^y value is = 25

(3.16) Write a program that read any number x and display e^x to the power x (e^x)
⇒ #include <stdio.h>
int main()

[C library function]

```
{ int x;
    double result;
    printf("Enter x : ");
    scanf("%d", &x);
    result = exp(x);
    printf("e^x value is = %f\n", result);
```

3

Output: Enter x : 2

e^x value is = 7.389056

$\ln(x)$

(3.17) Write a program that read any number x and display $\log(x)$

⇒ #include <stdio.h>

int main()

```
{ int x;
    double result;
```

[C library function]

```
    printf("Enter any number x : ");
    scanf("%d", &x);
```

```
    result = log(x);
```

```
    printf("log(%d) = %f\n", x, result);
```

3

Output: Enter any number : 5,

$\ln(5)$ = 1.609438

\ln നുംബർ ലൈൻ

അതു ക്രമാന്തരം

\ln നുംബർ ലൈൻ

3.18 Write a program that read any number x and display $\log_{10}(x)$.

```
#include <stdio.h>
int main()
```

```
{ int x;
double result;
printf ("Enter any number : ");
scanf ("%d", &x);
result = log10(x);
printf ("log10(%d) = %lf\n", x, result);
```

3
Output: Enter any number: 12
 $\log_{10}(12) = 1.07918$

X 3.2 Write a program that read any angle t and display $\sin(t)$.

```
#include <stdio.h>
#include <math.h>
int main()
```

```
{ float t, result; t_radians;
printf ("Enter any angle : ");
scanf ("%f", &t);
t_radians = t * (M_PI / 180);
result = sin(t_radians);
printf ("sin(%f) = %f\n", t, result);
```

3
Output: Enter any angle : 30
 $\sin(30) = 0.50$

$$t\text{-radians} = \left(t \times \frac{\pi}{180} \right);$$

$$result = \sin(t\text{-radians});$$

[We know]

$$\text{radians} = \text{degree} \times \frac{\pi}{180}$$

⑤ round function: ଏହି function ଏହା କାହାର ହଲୋ ଦକ୍ଷିଣାତ୍ମକ ପତ୍ର
କୁ ରୁକ୍ଷ୍ୟା ଥାଏଣ ଅ ଚାଇବାର ଜୀବି ହଲେ ଦକ୍ଷିଣାତ୍ମକ ଗଛେ ଏହି ରୁକ୍ଷ୍ୟାଟି
ଆରୁ ତାରେ ପ୍ରାକ୍ତନ ଲାଗା ଥାଏ ଯଦି ଦକ୍ଷିଣାତ୍ମକ ପତ୍ର ଚାହିଁ ତାହା ରୁକ୍ଷ୍ୟା
ଥାଏ ତାହାର ଦକ୍ଷିଣାତ୍ମକ ଗଛେ ମୂର୍ଖରୁକ୍ଷ୍ୟା ଆଏ ଏବେଳେ ହତେ ଯେବେ:

Code: #include <stdio.h>
int main()

```

    } double x = 4.5232; printf;
    printf = round(x);
    printf ("%lf\n", x, result);

```

ii) trunc function: এই function এতে ক্ষেত্র অন্তর্ভুক্ত করা নথি।

trunc(4.522092)

$$\text{Output} = 4,000$$

Code: #include <ntddio.h>

int main()

{ double x= 5.6210 , result ; } //line 11

result = triangle(x);

```
printf ("drumc (%.1f) = %.1f", x, result);
```

$$3 \quad \text{EIX} = (\text{EIX}_{\text{final}})^{\frac{1}{n}} \text{EIX}_{\text{initial}}$$

Output: $\text{trunc}(5.6210) = 5$

Ceil Function: Ceil function නේ තාවත් නොමැති ප්‍රක්‍රියා හෝ ප්‍රතිඵලීය ප්‍රක්‍රියා වෙති. Ceil (-2.012) නීතිය නොමැති නොවා ඇත.

$\text{ceil}(2.9012)$
output = 3

$$\text{output} = -2$$

Code: #include <stdio.h>
int main()

С - узлы, пепел;

2 double $x = 4.0124$, $r =$
 $\sqrt{171}$:

$\text{result} = \text{ceil}(x)$

$\text{percent} = \text{ceil}(\%df)$

```
printf("cell(%d)
```

Output: $\text{ceil}(4.0124) = 5$

floor function- $\lfloor x \rfloor$ Function ଏହି କାଣ୍ଡା
କିମ୍ବା ନିମ୍ନ ପରିମାଣ ଅଳ୍ଯଗାଣ ଜମାତା.

ପ୍ରକାଶନ ତାରିଖ (DD/MM/YY) ୧୫/୦୮/୨୦୨୩

7

example:

Ex 1

- floor (2.8)

Outfit: 1

Code: #include <stdio.h>

```
int main()
```

{ double $x = -3.402$, result }

пенальт = floor(n);

```
printf("floor(%lf) = %lf", x, result),
```

2

Output:

$$\text{floor}(-3.402) = -4$$

Assignment operation

Assignment operation	Example	Full meaning
\equiv	$y \equiv x + 5$	$y = x + 5$
$+=$	$x += 5$	$x = x + 5$
$-=$	$x -= y$	$x = x - y$
$*=$	$x *= 5$	$x = x * 5$
$/=$	$x /= 5$	$x = x / 5$
$\% =$	$x \% = 5$	$x = x \% 5$

Q) Code of Assignment operation.

$\Rightarrow \#include <stdio.h>$

int main()

int a = 5;

$a += 2;$

printf ("%d", a);

$a -= 2;$

printf ("%d", a);

$a *= 2;$

printf ("%d", a);

$a /= 2;$

printf ("%d", a);

$a \% = 2;$

printf ("%d", a);

3 printf ("%d", a);

Output:

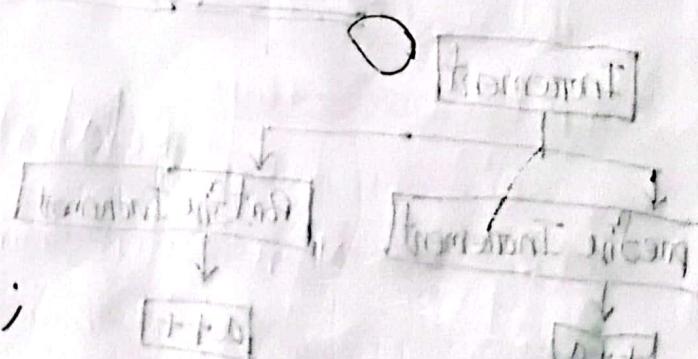
7

3

10

3

1



Unary operation

Unary operation	Meaning
+	Unary plus
-	Unary minus
++	Increment
--	Decrement

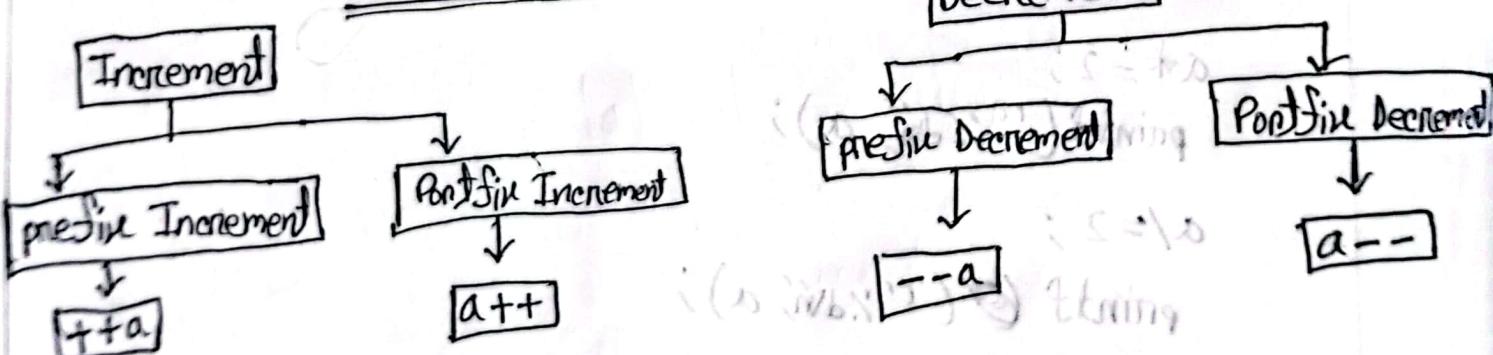
सिर्जना बान फिल्म एवं
एवं सिर्जना बान फिल्म एवं

Code: Unary plus
 $\Rightarrow \#include <stdio.h>$
 $\text{int main}()$
 $\{ \text{int } x=10;$
 $\text{int result} = +x;$
 $\text{printf} ("%d\n", result);$
 $\}$
Output: 10

Code: Unary minus

$\Rightarrow \#include <stdio.h>$
 $\text{int main}()$
 $\{ \text{int } x=10;$
 $\text{int result} = -x;$
 $\text{printf} ("%d\n", result);$
 $\}$
Output: -10.

Increment and Decrement operator



increment operator:

$++x$; // increments x by one - Before it is used
 $x++$; // increments x by one - After it is used

decrement operator:

$--x$; // decrements x by one - Before it is used
 $x--$; // decrements x by one - After it is used

④ Use of postfix Increment

⇒ #include <stdio.h>

int main()

{ int x=10;

int y=x++;

printf ("x=%d\n", x);
printf ("y=%d\n", y);

}

Output: x=10
y=10

⑤ Use of prefix increment

⇒ #include <stdio.h>

int main()

{ int x=20;

int y=++x; //y=9

printf ("x=%d\n", x); //x=9

printf ("y=%d\n", y); //y=9

}

Output: x=9
y=9

Postfix operator

⑥ Use of prefix Increment

⇒ #include <stdio.h>

int main()

{ int x=10;

int y=++x;

printf ("x=%d\n", x);
printf ("y=%d\n", y);

Output: x=11
y=11

⑦ Use of postfix decrement

⇒ #include <stdio.h>

int main()

{ int x=20;

int y=x-- //y=10

printf ("x=%d\n", x); //x=19

printf ("y=%d\n", y); //y=10

}

Output: x=9
y=10

Prefix notation result ← <

lions no result ← = <

North south ← >

lions no result east ← = >

lions lions ← = =

lions tall ← = !

② Use of unary operators

④ Line 65
⇒ #include <ntdio.h>

```
int main()
```

```
{ int n=20;  
    printf ("%d\n", x++);  
    printf ("%d\n", n);  
    printf ("%d\n", ++x);  
    printf ("%d\n", x);  
    printf ("%d\n", n--);  
    printf ("%d\n", --x);
```

Autodid: 10

$$11x = 10$$

$$j(n) = \frac{1}{4} b^n = 3^{(n)} \text{ Eddington}$$

$$\mu = \left(\frac{1}{b^2} + b^2 \right)^{-\frac{1}{2}}$$

$$\mu u = 1^2$$

$$1111 = 12$$

$$11u =$$

$$11 \quad n = 10$$

Relational operators

$>$ → Mean greater than

\geq → greater than or equal

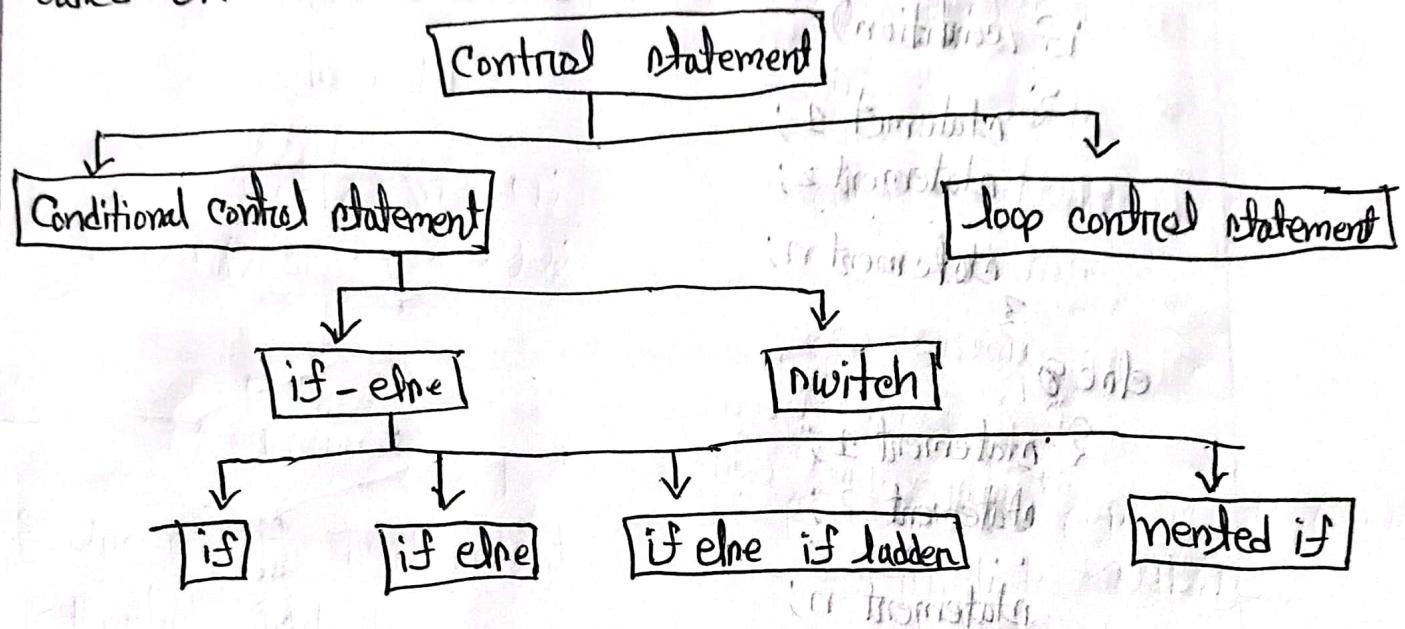
< → leare than

\leq \rightarrow less than or equal

\equiv \equiv \rightarrow equal equal

\neq → not equal

Control statement of C: You have made some decisions based on some conditions



Q.1 Write a program that reads an integer and prints odd or even.

⇒ #include<stdio.h>

```
int main()
{
    int num;
    printf("Enter any number :");
    scanf("%d", &num);
    If (num%2 == 0)
        printf ("Even\n");
    else
        printf ("Odd\n");
}
```

Output: Enter any number : 20

Even .

if - eine Syntax:

```

graph TD
    A["if (condition)"] --> B["statement 1;  
statement 2;  
statement n;"]
    A --> C["else {  
    statement 1;  
    statement 2;  
    statement n;}"
    ]
    B --> D["aus - ti"]
    C --> E["nicht - ti"]
    E --> F["aus - ti"]
    F --> G["nicht - ti"]

```

Ergebnis: Lernst du Lernst du
Lernst du nicht nicht

3

4.2 - 4.5 Write program that read two numbers and display maximum and minimum number.

⇒ #include <stdio.h>

```

int main()
{
    int num1, num2;
    printf ("Enter 1st number = ");
    scanf ("%d", &num1);
    printf ("Enter 2nd number = ");
    scanf ("%d", &num2);
    if (num1 > num2)
    {
        printf ("Maximum number is = %d\n", num1);
        printf ("Minimum number is = %d\n", num2);
    }
}

```

• Error and wrapper: digit

```

else if (num1 > num2)
{
    printf ("Maximum number is = %d\n", num1);
    printf ("Minimum number is = %d\n", num2);
}
else
    printf ("Number are equal");
}

```

Output: Enter 1st number = 62

Enter 2nd number = 94

Maximum number is = 94

Minimum number is = 62

(1.1) Write a program that read mark and display result in grade.

→ #include <stdio.h>

```

int main ()
{
    float mark;
    printf ("Enter your mark : ");
    scanf ("%f", &mark);
    if (mark >= 80)
        printf ("A\n");
    else if (mark >= 70)
        printf ("A-\n");
    else if (mark >= 60)
        printf ("B+\n");
    else if (mark >= 50)
        printf ("B\n");
    else if (mark >= 40)
        printf ("C\n");
    else if (mark >= 33)
        printf ("D\n");
    else
        printf ("Fail\n");
}

```

Output: Enter your mark : 78.5

A

④ Write a program that algorithm, flowchart and program that read any number and display Even or odd.

→ Algorithm

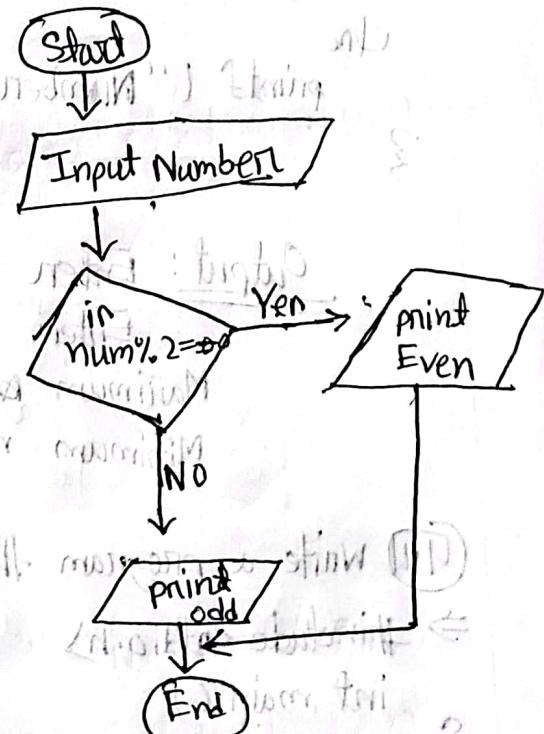
- ① Start
- ② Input number
- ③ If $\text{num} \% 2 == 0$
 - i) Yes, print Even
 - ii) False, print Odd
- ④ End

Program :

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

Output : Enter any number : 35
The number is odd.

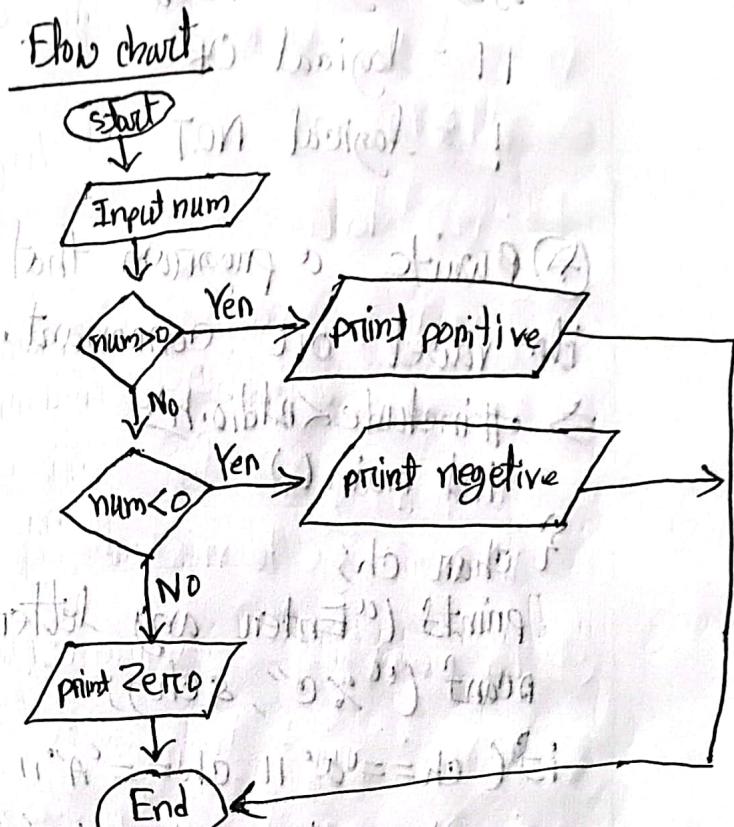
Flow chart



④ Write a Algorithm, flow chart and program that read and numbers and define thin numbers in positive or negative

⇒ Algorithm

- ① Start
- ② Input thin number
- ③ If $\text{num} > 0$
 - i) Yes, print positive
 - ii) No, go to next step
- ④ If $\text{num} < 0$
 - i) Yes, print negative
 - ii) No, print zero
- ⑤ End



Program

```
#include <stdio.h>
int main()
{
    int num;
    printf ("Enter any number : ");
    scanf ("%d", &num);
    if (num > 0)
        printf ("The number is positive");
    else if (num < 0)
        printf ("The number is negative");
    else
        printf ("The number is zero");
```

Output: Enter any number : -32

The number is negative

Logical operation

& = logical AND

|| = logical OR

! = logical NOT

Write a program that read any character and display it vowel or consonant.

#include <stdio.h>

int main()

{ char ch;

printf ("Enter any letter :");

scanf ("%c", &ch);

if (ch == 'a' || ch == 'A' || ch == 'e' || ch == 'E' ||

ch == 'i' || ch == 'I' || ch == 'o' || ch == 'O' ||

ch == 'u' || ch == 'U')

printf ("The letter is vowel.");

else

3 printf ("The letter is consonant.");

Output: Enter any letter : G

The letter is consonant.

Q) Write a program that read 3 numbers and display largest number.

Code: `#include <stdio.h>`

```
int main()
{
    int num1, num2, num3;
    printf("Enter 3 numbers : ");
    scanf("%d %d %d", &num1, &num2, &num3);
    if (num2 > num1 && num2 > num3)
        printf("Largest number is = %d\n", num2);
    else if (num1 > num2 && num1 > num3)
        printf("Largest number is = %d\n", num1);
    else if (num3 > num1 && num3 > num2)
        printf("Largest number is = %d\n", num3);
    else
        printf("Numbers are equal");
}
```

Output: Enter 3 Numbers: 50, 40, 65

Largest number is = 65.

Q) Write a program that read 3 numbers and display minimum number.

Code:

```
#include <stdio.h>
int main()
```

```
{ int num1, num2, num3;
```

Largest number is = 65
Minimum number is = 40

④ Write algorithm, flow chart and C program that read a year and display its leap year or not leap year.

⇒ Algorithm

Step: ① Start

② Input a year

③ If $\text{year} \% 400 == 0$

i) Yes; print leap year

ii) No; go to next step in condition

④ If $\text{year} \% 4 == 0$ and $\text{year} \% 100 != 0$

i) Yes; print leap year

ii) No, print Not leap year

⑤ End

Code: ~~#include <stdio.h>~~

~~int main()~~

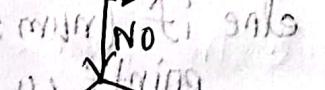
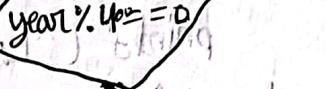
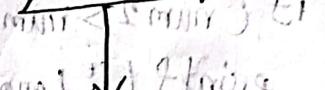
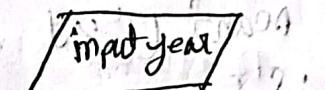
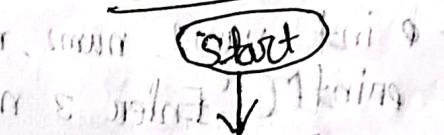
~~printf("Enter a year:");~~

~~scanf("%d", &year);~~

~~if (year % 400 == 0)~~

~~printf("Leap year?");~~

flowchart



Code: #include <stdio.h>

int main()

{

 int year;

 printf("Enter a year:");

 scanf("%d", &year);

 if (year % 400 == 0)

 printf("Leap year?");

```

else if (year % 4 == 0 && year % 100 != 0),
    printf ("Leap year");
else
    printf ("Not leap year");
}

```

Output: Enter a year : 2024
Leap year.

④ Write a algorithm, flow chart and C program that read a character and display it in small or capital.

⇒ Algorithm.

- ① Start
- ② Input character

③ If $ch \geq A \text{ and } ch \leq Z$

(i) Yes ; print capital

(ii) No ; go to next step

④ If $ch \geq a \text{ and } ch \leq z$

(i) Yes ; print small

(ii) No ; print Not a letter

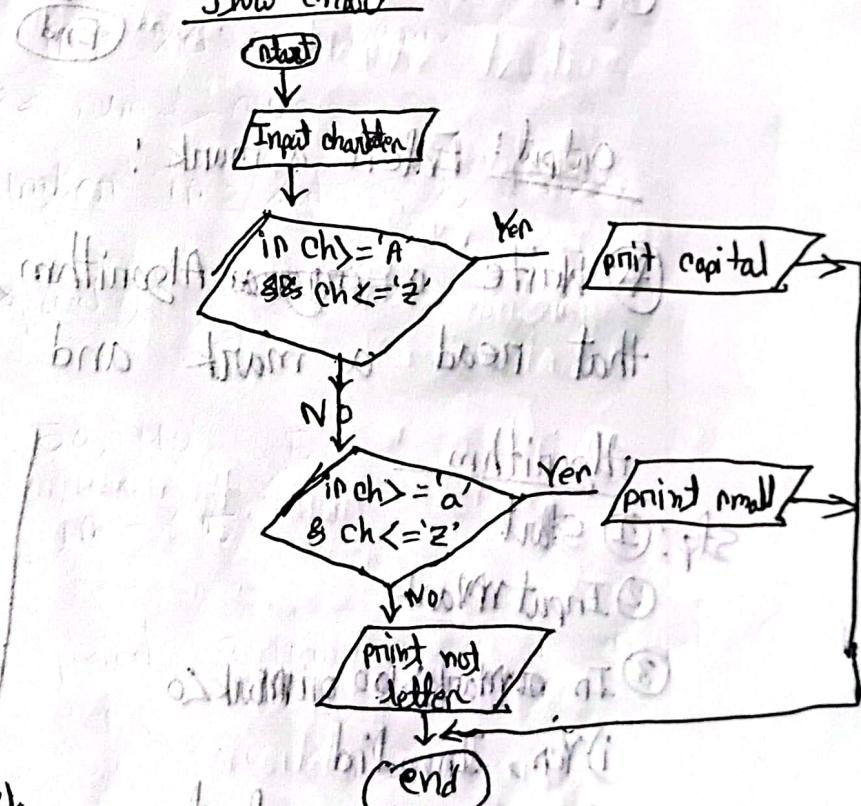
Code: #include <stdio.h>

```

int main()
{
    char ch;
    printf ("Enter any character : ");
    scanf ("%c", &ch);
    if (ch >= 'A' && ch <= 'Z')
        printf ("Capital letter");
    else if (ch >= 'a' && ch <= 'z')
        printf ("Small letter");
    else
        printf ("Not letter");
}

```

Flow chart



Output: Enter any character: H
Capital letter

Algorithm, flow chart and program that read a mark and display pass or fail.

⇒ Algorithm:

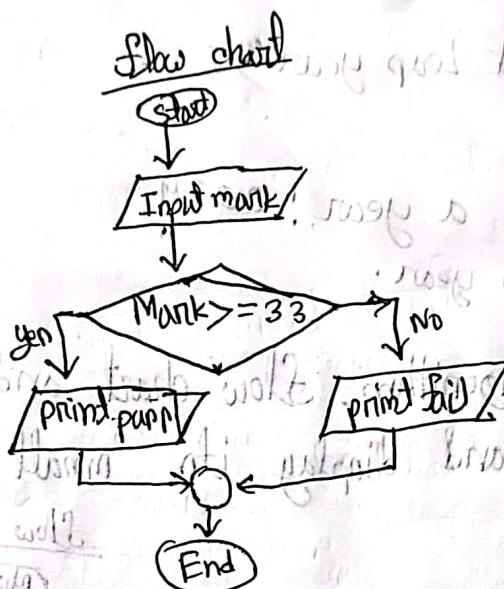
Step ①: Start

② Input mark

③ If $mark >= 33$

- i) Yes, print pass
- ii) No, print fail

④ End



Output: Enter a mark :

Write a ~~program~~ algorithm, flow chart and program that read a mark and display letter of grade.

Algorithm

Step ① Start

② Input mark

③ If ~~mark > 100 or mark < 0~~

i) Yes, Invalid

ii) No, go to next step.

④ If $mark \leq 100$ and $mark \geq 80$

i) Yes, print "A+"

ii) No, go to next step.

⑤ If $mark \leq 79$ and $mark \geq 70$

i) Yes, print 'A'

ii) No, go to next step

⑥ If $mark \leq 69$ and $mark \geq 60$

- i) Yes, print 'B'
- ii) No, go to next step

⑦ If $mark \leq 59$ and $mark \geq 50$

- i) Yes, print 'C'
- ii) No, go to next step

⑧ If $mark \leq 49$ and $mark \geq 40$

- i) Yes, print 'D'
- ii) No, go to next step

⑨ If $mark \leq 39$ and $mark \geq 33$

- i) Yes, print 'D'
- ii) No, print fail

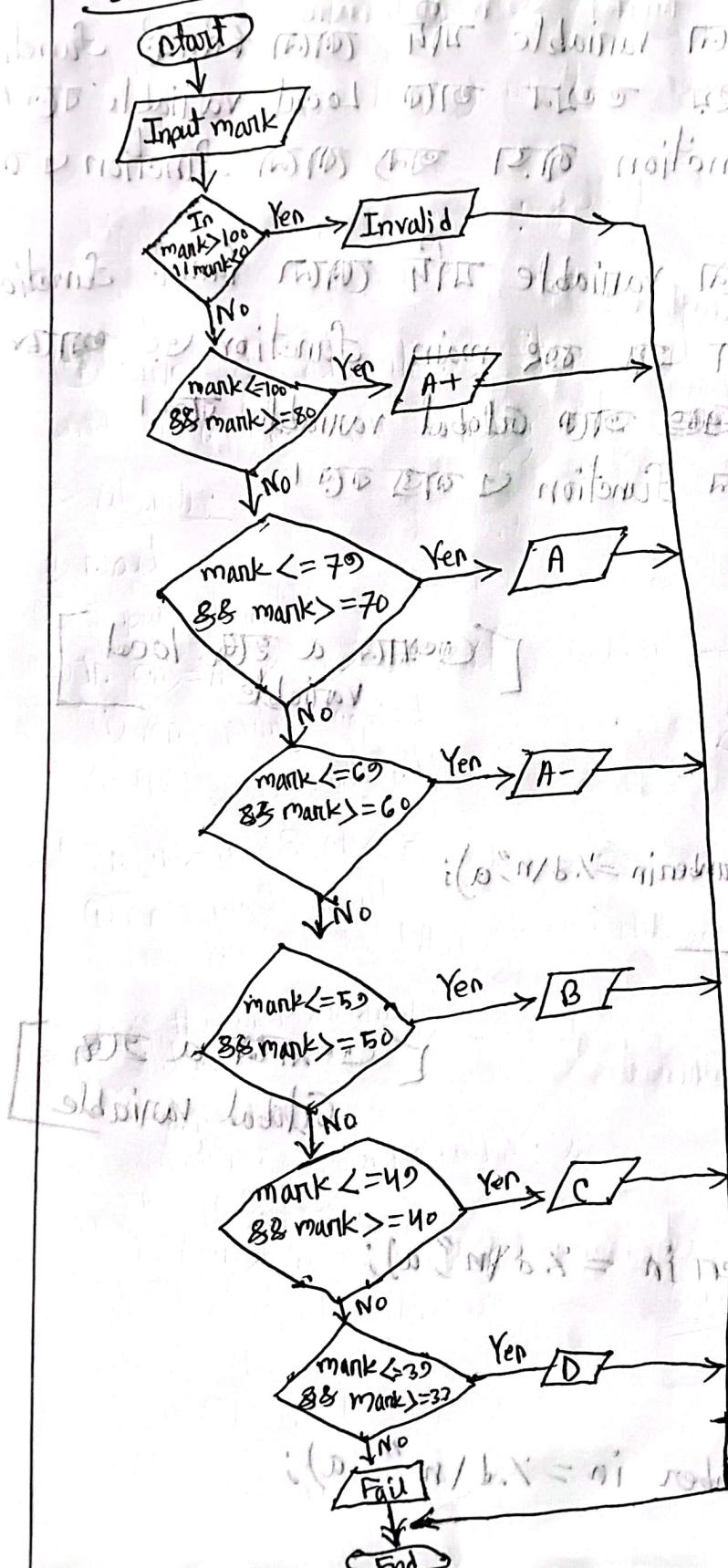
10 End

Code:

```
#include <stdio.h>
int main()
{
    float mark;
    printf("Enter a mark:");
    scanf("%f", &mark);
    if (mark >= 33)
        printf("pass");
    else
        printf("fail");
}
```

Flow chart

Code:



```

#include <stdio.h>
int main() {
    float mark;
    printf("Enter a mark : ");
    scanf("%f", &mark);
    if (mark > 100 || mark < 0)
        printf("Invalid");
    else if (mark <= 100 & mark >= 80)
        printf("A+");
    else if (mark <= 79 & mark >= 70)
        printf("A");
    else if (mark <= 69 & mark >= 60)
        printf("A-");
    else if (mark <= 59 & mark >= 50)
        printf("B");
    else if (mark <= 49 & mark >= 40)
        printf("C");
    else if (mark <= 39 & mark >= 32)
        printf("D");
    else
        printf("Fail");
}
  
```

Output: Enter a mark: 78.5

* Local variable and Global variable

⇒ Local variable: কোনো variable যদি কোনো নির্দিষ্ট function

o declaration করা হয় তখন তাকে Local variable বলে।

o local variable এর function কিম্বা অন্য কোনো function এ কাজ

করতে পারে।

⇒ Global variable: কোনো variable যদি কোনো নির্দিষ্ট function

o declaration করা না হয় তবে main function এ কাজ

declaration করা হয় তাকে Global variable বলে।

o Global variable কোনো function এ কাজ করে।

Example:

// Local variable

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

```
int main()
```

```
{ int a = 5;
```

```
printf ("Include the number in = %.d\n", a);
```

```
}
```

// Global variable

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

```
int a = 5;
```

```
int main()
```

```
{ printf ("The number in = %.d\n", a);
```

```
display();
```

```
Void display()
```

```
{ printf ("The number in = %.d\n", a);
```

```
}
```

[গোচার a হচ্ছে local variable]

[গোচার a হচ্ছে Global variable]

Switch statement

- ④ Switch (~~do~~) up to keyword
switch, case, break, default.

- ④ Write a program that read a digit and display its spelling.

```
#include <stdio.h>
int main()
```

```
{ int digit;
printf ("Enter a digit :");
scanf ("%d", &digit);
Switch (digit)
{
```

```
    Case 0 :
        printf ("Zero\n"); break;
```

```
    Case 1 :
        printf ("One\n"); break;
```

```
    Case 2 :
        printf ("Two\n"); break;
```

```
    Case 3 :
        printf ("Three\n"); break;
```

```
    Case 4 :
        printf ("Four\n"); break;
```

```
    Case 5 :
        printf ("Five\n"); break;
```

```
    Case 6 :
        printf ("Six\n"); break;
```

```
    Case 7 :
        printf ("Seven\n"); break;
```

```
    Case 8 :
        printf ("Eight\n"); break;
```

```
    Case 9 :
        printf ("Nine\n"); break;
```

```
    default printf ("Not valid");
```

Output : ~~Enter~~
Enter a digit : 8

Eight

Q) Write a program that read a character and display vowel or consonant with switch case statement.

⇒ Code:

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

```
int main()
```

```
{
```

```
    char ch;
```

```
    printf("Enter any character : ");
```

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

```
    switch(ch)
```

```
{
```

```
        Case 'a':
```

```
        Case 'e':
```

```
        Case 'i':
```

```
        Case 'o':
```

```
        Case 'u':
```

```
        Case 'A':
```

```
        Case 'E':
```

```
        Case 'I':
```

```
        Case 'O':
```

```
        Case 'U':
```

```
        printf("Vowel");
```

```
        break;
```

```
    default:
```

```
        printf("Consonant");
```

3

Output: Enter a character: G

: Consonant

: Abcd

: EFGH

: IJKL

: MNOP

④ Write a C program that reads ~~temp~~ Fahrenheit or Celsius temperature and display Celsius or Fahrenheit temperature with switch statement.

⇒ Code:

```
#include <stdio.h>
int main()
{
    int choice;
    float temp;
    printf("Temperature conversion menu\n");
    printf("1. Fahrenheit to Celsius\n");
    printf("2. Celsius to Fahrenheit\n");
    printf("Enter your choice");
    scanf("%d", &choice);
    switch (choice)
    {
        case(1):
            {
                printf("Enter the Fahrenheit temperature");
                scanf("%f", &temp);
                result = (temp - 32) / 1.8;
                printf("The temperature in Celsius is %.f\n", result);
            }
        case(2):
            {
                printf("Enter the Celsius temperature");
                scanf("%f", &temp);
                result = (temp * 1.8) + 32;
                printf("The temperature in Fahrenheit is %.f\n", result);
            }
    }
}
```

We know -

$$\frac{F}{5} = \frac{F - 32}{9}$$

$$\therefore C = \frac{F - 32}{1.8}$$

$$\text{and } \frac{F}{5} = \frac{C \times 1.8 + 32}{9}$$

$$\therefore F = (C \times 1.8) + 32$$

?

?

(brackets) solution

?

?

?

?

?

?

?

?

?

?

?

?

?

?

?

?

?

?

Output: Temperature conversion menu

- ① Fahrenheit to Celsius
- ② Celsius to Fahrenheit

Enter your choice

1

Enter your Fahrenheit temperature : 98

The temperature in Celsius is = 36.67

④ Write a C program that read a password and check the password.

$\Rightarrow \#include <stdio.h>$

int main()

{ int password;

printf("Enter any password");

scanf("%d", &password);

switch(password)

{

case 1122: printf("Correct password"); break;

default: printf("Incorrect password"); break;

case 1122: printf("Correct password"); break;

default: printf("Incorrect password"); break;

}

}

3

Output: Enter any password : 1122

Correct password.

④ Write a C program that read month number and display month Name.

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

```
int main()
```

```
{ int num;
```

```
printf("Enter month number:");
```

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

```
switch (num)
```

```
{ Case (1):
```

```
    printf("January");
```

```
break;
```

```
Case (2):
```

```
    printf("February");
```

```
break;
```

```
Case (3):
```

```
    printf("March");
```

```
break;
```

```
Case (4):
```

```
    printf("April");
```

```
break;
```

```
Case (5):
```

```
    printf("May"); break;
```

```
Case (6):
```

```
    printf("June"); break;
```

```
Case (7):
```

```
    printf("July"); break;
```

```
Case (8):
```

```
    printf("August"); break;
```

```
Case (9):
```

```
    printf("September"); break;
```

```
Case (10):
```

```
    printf("October"); break;
```

```
Case (11):
```

```
    printf("November"); break;
```

```
Case (12):
```

```
    printf("December"); break;
```

```
default:
```

```
    printf("Not valid");
```

3
Output: Enter month number: 22

December.

Conditional operations

? :

Write a C program that read two numbers and display large number with conditional operation.

⇒ #include <stdio.h>

```

int main()
{
    int num1, num2;
    printf("Enter two numbers : ");
    scanf("%d %d", &num1, &num2);
    large = (num1 > num2) ? num1 : num2;
    printf("Large number is = %d\n", large);
}

```

Output: Enter two numbers : 5, 10

Large number is = 10

Bitwise Operator

⇒ Bitwise operator ରାଖିଲୁଣି ଏହା ଅର୍ଥାତ୍ Bit / Byte ବିଷେ କାହା କାହା

ଏହା କାହାରୁ ନିଜିକୁ ପିଛି ଦେଖିଲୁ ଅପାରେଟର ଅମ୍ବଳ କହା ହେବା।
ଏହା - AND, OR, NOT, EXOR, Left shift, Right shift etc.

[It can only used on integer, don't work with float]

Bitwise operator	Meaning
&	Bitwise AND
	Bitwise OR
^	Bitwise EXOR (XOR)
>>	Right shift
<<	Left shift
~	Bitwise NOT

★ Bitwise program:

```
#include <stdio.h>
int main()
{
    int a=32;
    int b=12;
    int c;
    c=a&b;
    printf("a&b=%d\n", c);
    c=a|b;
    printf("a|b=%d\n", c);
}
```

```
c=a^b;
printf("a^b=%d\n", c);
```

Output:

$$\begin{aligned} a \& b &= 0 \\ a | b &= 44 \\ a ^ b &= 44 \end{aligned}$$

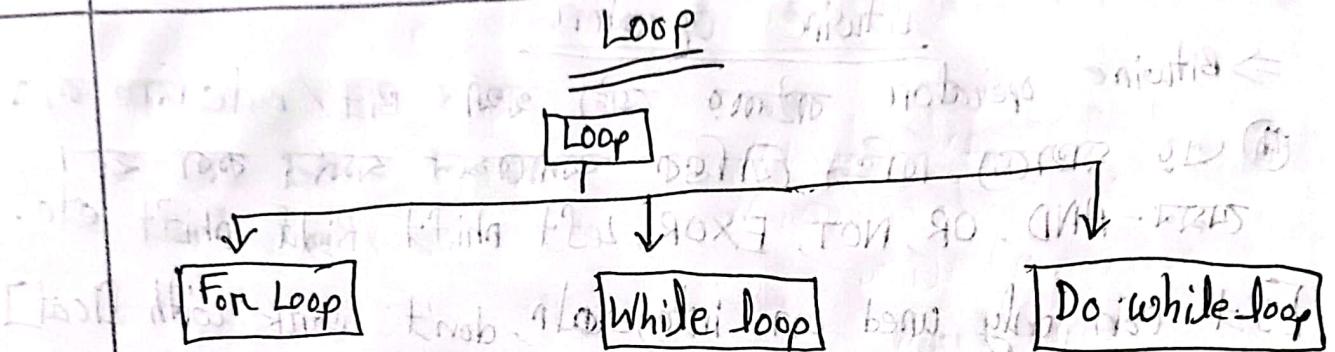
$$\begin{array}{r} a = 32 = 00100000 \\ b = 12 = 00001100 \\ \hline a \& b = 00000000 \end{array}$$

$$\begin{array}{r} a = 32 = 00100000 \\ b = 12 = 00001100 \\ \hline a | b = 00101100 = 44 \end{array}$$

$$\begin{array}{r} a = 32 = 00100000 \\ b = 12 = 00001100 \\ \hline a ^ b = 00101100 = 44 \end{array}$$

EXOR କିମ୍ବା କାହାରୁ (Same input = Output 0)

:(najibhmo) shuktara



For loop syntax:

Counter declaration;

for (initialization; condition; update)

{
 Loop body
}

Update mean increment
or decrement

while loop syntax:

declaration;

initialization;

while (condition)

{
 update
}

do-while loop syntax:

declaration; $i = 0$

initialization; $i > 0$

($i > 0$) do {
 ~~i = i - 1~~
 ~~cout << i~~
 update
}

 while (condition);

Hint 2

$$i = 10$$

$$i = 10$$

$$i = 10$$

Write a program that print "Belayet Honnain" 5 times.
 $\Rightarrow \#include <stdio.h>$ (One for Loop)

```

int main()
{
    int i;
    for (i=1; i<=5; i++)
    {
        printf("i=%d\n", i);
        printf("Belayet Honnain");
    }
    return 0;
}
  
```

Output:

i = 1	i = 5
Belayet Honnain	Belayet Honnain
i = 2	
Belayet Honnain	
i = 3	
Belayet Honnain	
i = 4	
Belayet Honnain	

i = 5
 Belayet Honnain

Use while loop

④ Write a program that print "Belayet Honnain" 5 times

using while loop.

⇒ #include <stdio.h>
int main()

{ int i=1;
while (i<=5)

{ printf ("i=%d\n", i);
printf ("Belayet Honnain");

i++;

"Minnot föld" string titt maring a stink
return 0;

}

Output: i=2

Belayet Honnain.

i=2

Belayet Honnain.

i=3

Belayet Honnain "minnot föld" string

i=4

Belayet Honnain.

i=5

Belayet Honnain. i

minnot föld

// while loop
// initialization;
while (condition)
 { counterupdate
 // }

"Minnot föld" string titt maring a stink
(good not bad) <stink> skuffit ←
(smell tri)

i tri ?

(++i : i>i if =i) not

(i, "m/bx = i") string }

:0 nnter

s = i : hñtio

s = i Belayet Honnain

s = i minnot föld

n = i minnot föld

go back to intro work

Use Do while loop to intro work

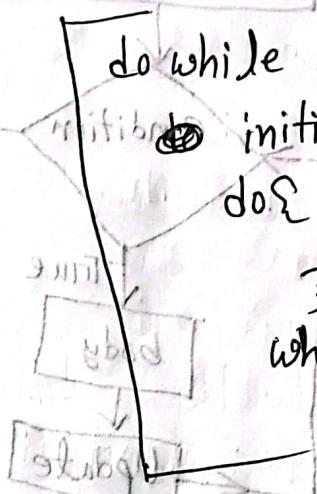
- ④ Write a program that print "Belayet Honnain" 3 times using do while loop.

⇒ #include <stdio.h>

int main()

```
{ int i=1; // initialization  
do {  
    printf("i=%d\n", i);  
    printf("Belayet Honnain");  
    i++; // increment;  
}  
while (i<=3); // condition;  
return 0;  
}
```

motivasi kintini



do while loop consists of

initialization;

do { increment or decrement;

3

while (condition);

Output: i=1

Belayet Honnain

i=2

Belayet Honnain

i=3

Belayet Honnain

motivasi kintini

- ④ Write a program that print even numbers 2 to 10.

⇒ #include <stdio.h>

int main()

```
{ int i;
```

```
for (i=2; i<=10; i=i+2)
```

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

```
return 0;
```

```
}
```

Output:

i=2

i=4

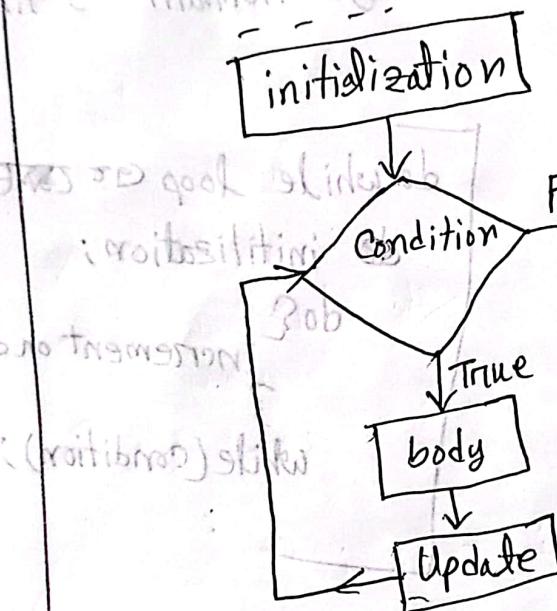
i=6

i=8

i=10

Flow chart of loop

Flow chart of for loop:



Flow chart of while loop:

initialization

Condition

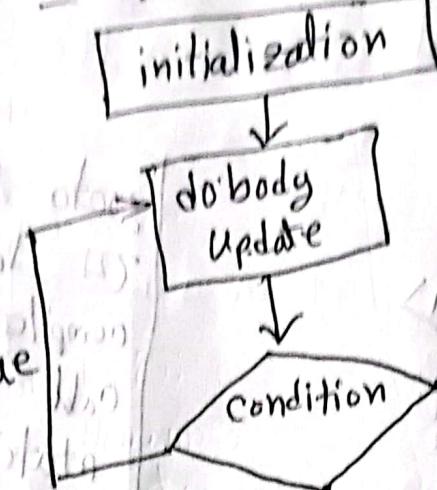
body
Update

False

Exit

$s = i$
 $N = i$
 $l = i$
 $8 = i$
 $oP = i$

Flow chart of Do-while Loop:



break and continue

```

④ #include <stdio.h>
int main()
{
    int i;
    for( i=1; i<=100; i++)
    {
        if( i%3 == 0)
            continue;
        printf(" i=%d\n", i);
        if( i== 10)
            break;
    }
}
  
```

Output

```

i = 1
i = 2
i = 4
i = 5
i = 7
i = 8
i = 10
  
```

"Continue" keyword പോക്കാൽ
ഇല്ല by pass കരി യാഥാം
നിങ്ക് എ ഫിയേ ആവായാം Loop
എ ഫിനേ യാഥാം | അന്തിമം,

"break" keyword പോക്കാൽ
ഒരു Loop കു ഓഫീസൈ ആജു
യാഥാം terminating പോ
ക്കു ചുറു

go to keyword

→ goto label; [Structure of goto statement]

label :

Statement n

⦿ C program :

```
#include <stdio.h>
int main()
```

```
{ int i=1;
```

beloyet :

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

```
i++;
```

```
if ( i<=5)
```

```
goto beloyet;
```

```
return 0;
```

Output: i=1

প্রেসেক্ট
label

no possibility

goback
label

condition

goto keyword go করে
label থেকে যাবে
compiler এর label করে
call করে যাবে যাবে
statement কুন্তা read করে
গো করে অনুযায়ী করে যাবে

প্রোগ্রাম প্রিন্ট করি ফাঁকাখানা
অনুসৃত করে নির্দেশ প্রদান করে
অর্থাৎ tab দিয়ার ক্ষেত্রে প্রিন্ট করা হয়

(i=3 : 002 : i=4 : 003 : i=5 : 004 :)

(0 == EX i=1 : 003 :)
(i == 1 : 003 :)

(0 == i) ? i : word

twink

L = i
S = i
H = i
R = i
P = i
B = i
O = i

④ Write a c program that read any number and display thin number multiplication tabel.

⇒ Code:

```
#include <stdio.h>
int main()
{
    while(1)
    {
        int n, i;
        printf("Enter any number : ");
        scanf("%d", &n);
        for(i=1; i<=10; i++)
        {
            printf("%d X %d = %d\n", n, i, n*i);
        }
    }
    return 0;
}
```

Output: Enter any number : 4

$$\begin{aligned} 4 \times 1 &= 4 \\ 4 \times 2 &= 8 \\ 4 \times 3 &= 12 \\ 4 \times 4 &= 16 \\ 4 \times 5 &= 20 \\ 4 \times 6 &= 24 \\ 4 \times 7 &= 28 \\ 4 \times 8 &= 32 \\ 4 \times 9 &= 36 \\ 4 \times 10 &= 40 \end{aligned}$$

Enter any number : 5

$$\begin{aligned} 5 \times 1 &= 5 \\ 5 \times 2 &= 10 \\ 5 \times 3 &= 15 \\ 5 \times 4 &= 20 \\ 5 \times 5 &= 25 \\ 5 \times 6 &= 30 \\ 5 \times 7 &= 35 \\ 5 \times 8 &= 40 \\ 5 \times 9 &= 45 \\ 5 \times 10 &= 50 \end{aligned}$$

प्रोग्राम while(1) का
दिलेंड प्रोग्राम का कार्य
करता किंतु ये नाम्बरों
में से हमें किसी भी
multiplication tabel का
नामता देखाता। फिर
while(1) loop जारी
करता किंतु यह यूनिफ
क्यूंधरा भी बताया
गया एवं एसें
नामता भी कहा याते

Q) Write an algorithm, flow chart and C program that read a number and display its factorial.

→ Algorithm

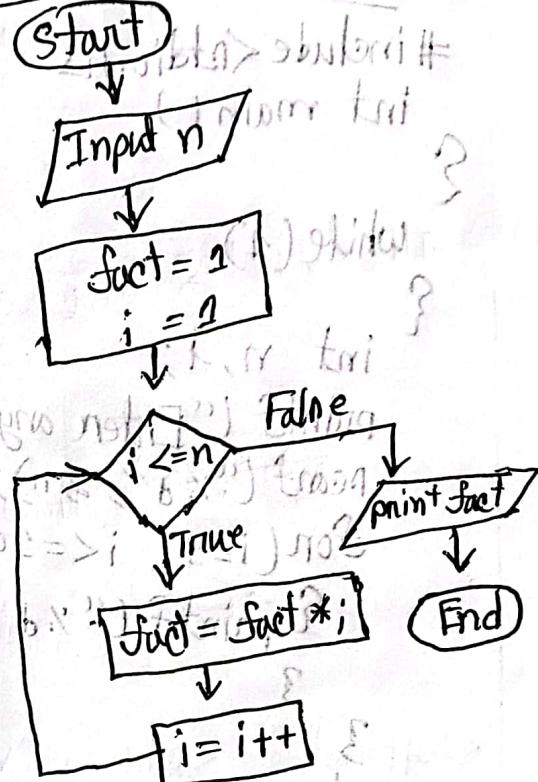
- ① Start
- ② Input n
- ③ fact = 1
 $i = 1$
- ④ while $i \leq n$
repeat step 5
- ⑤ fact = fact * i
- ⑥ print fact
- ⑦ End

C program:

```
#include <stdio.h>
int main()
{
    int n, i, fact = 1;
    printf("Enter any number : ");
    scanf("%d", &n);
    for (i=1; i<=n, i++)
    {
        fact = fact * i;
    }
    printf("Factorial = %d\n", fact);
    return 0;
}
```

Output: Enter any number: 5
Factorial = 120

Flow chart



① Write a Algorithm, Flow chart and C program that read a number and display it's prime or not prime.

⇒ Algorithm: program

```
#include <stdio.h>
int main()
```

```
{ int num, count = 0, i;
printf ("Enter any positive number\n");
scanf ("%d", &num);
for (i=2; i<num; i++)
```

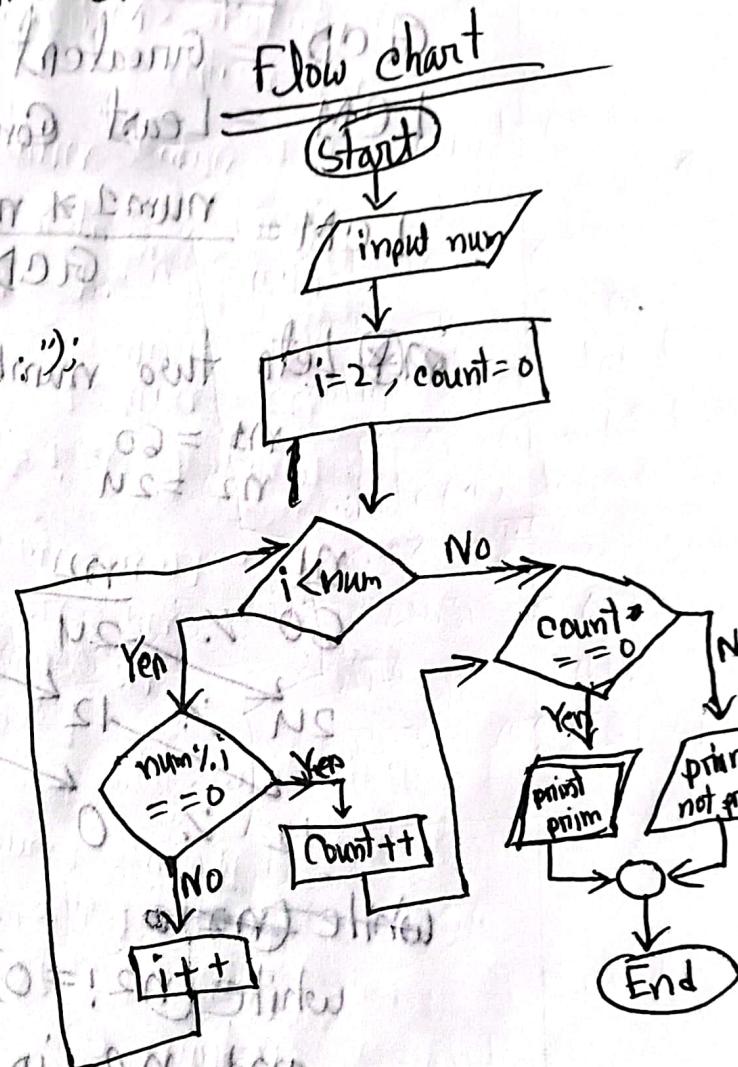
```
{
    if (num % i == 0)
        {
            count++;
            break;
        }
}
```

```
if (count == 0)
    printf ("prime number");
else
    printf ("Not prime number");
}
```

Algorithm:

- ① Start
- ② Input num
- ③ $i=2, \text{Count}=0$
- ④ If $i < \text{num}$
 - ① Yes, go to step 5
 - ② No, go to step 6

- ⑤ In, $\text{num} \% i == 0$
 - ① Yes, count++; go to step 6
 - ② No, $i++$; go to step 4
- ⑥ In, $\text{Count} == 0$.
 - ① Yes, print prime number
 - ② No, print not prime number
- ⑦ End



Find GCD & And LCM

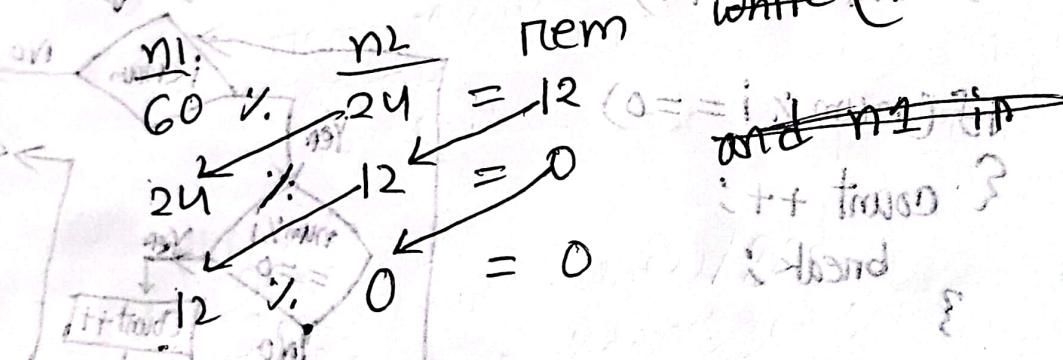
GCD = Greatest Common Divisor

LCM = Least Common Multiple

$$LCM = \frac{\text{num1} * \text{num2}}{GCD}$$

Q Let two numbers are 60 and 24. Find into GCD

$$\begin{aligned} n_1 &= 60 \\ n_2 &= 24 \end{aligned}$$



white ~~(n2 != 0)~~
while ~~(n2 != 0)~~

and n2 is in GCD

2 gata ot 60 : i++ times . n1 (i)

N gata ot 60 : i++ times . n1 (i)

0 == i * n1 . n1 (i)

maximum summing . n1 (i)

maximum summing for living . n1 (i)

affin of it

brote (i)

min sum . n1 (i)

0 == i * n1 . n1 (i)

min > i . n1 (i)

① Write Algorithm, flow chart and programming that read two integer numbers and display its GCD and LCM

⇒

Algorithm

- ① Start
- ② set $n_1 = \text{num}^1$
 $n_2 = \text{num}^2$
- ③ Input num^1 num^2
- ④ set $n_1 = \text{num}^1$
 $n_2 = \text{num}^2$
- ⑤ while ($n_2 \neq 0$)

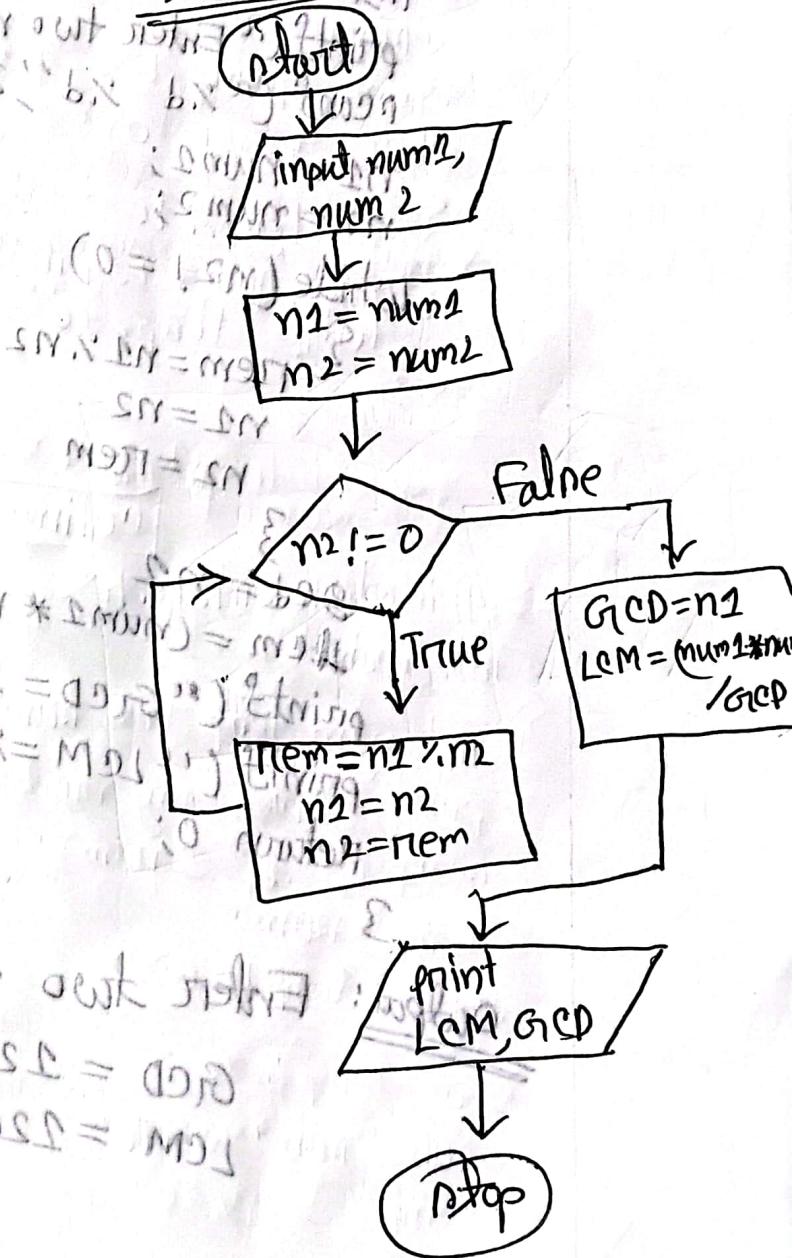
$$\left\{ \begin{array}{l} \text{rem} = n_1 \% n_2 \\ n_1 = n_2 \\ n_2 = \text{rem} \end{array} \right.$$

3
⑤ $\text{GCD} = n_2$
 $\text{LCM} = (\text{num}^1 * \text{num}^2) / \text{GCD}$

⑥ print GCD, LCM

⑦ Stop

Flowchart



C programming

```
#include <stdio.h>
int main()
```

```
{ int num1, num2, n1, n2, rem, gcd, lcm;
```

printf ("Enter two numbers : ");

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

```
n1 = num1;
n2 = num2;
```

```
while (n2 != 0)
```

```
    { rem = n1 % n2
```

```
        n1 = n2
```

```
        n2 = rem
```

```
    }
```

```
    gcd = n2
```

```
    lcm = (num1 * num2) / gcd
```

```
    printf ("GCD = %d\n", gcd);
```

```
    printf ("LCM = %d\n", lcm);
```

```
return 0;
```

Output: Enter two numbers = 60 24

GCD = 12

LCM = 120

* Write a program that read any positive integer and display sum of digits.

$$\Rightarrow 123 \\ = 1+2+3$$

theory

$$\begin{array}{r} \text{temp} \\ 10 | 123 \\ \underline{10} \quad | 23 \\ 2 \quad 0 \\ \underline{\underline{2}} \quad 3 \\ \quad \quad \quad \boxed{3} \end{array}$$

$$\begin{array}{r} \text{temp} \\ 10 | 12 \\ \underline{10} \quad | 2 \\ \quad \quad \quad \boxed{2} \end{array}$$

$$\begin{array}{r} \text{temp} \\ 10 | 1 \\ \underline{10} \quad | 0 \\ \quad \quad \quad \boxed{1} \end{array}$$

$$\begin{aligned} \text{Sum} &= 0 \\ \text{num} &= 123 \\ \text{temp} &= \text{num} = 123 \\ \text{while}(\text{temp} \neq 0) \\ \{ \quad r &= \text{temp} \% 10 \\ \text{num} &= \text{num} + r \\ \text{temp} &= \text{temp} / 10 \\ & \quad \quad \quad 3 \end{aligned}$$

Coding

```
#include <stdio.h>
int main()
{
    int num, temp, r, sum=0;
    printf("Enter any number : ");
    scanf("%d", &num);
    temp = num;
    while (temp != 0)
    {
        r = temp % 10;
        sum = sum + r;
        temp = temp / 10;
    }
    printf("Sum of digits : %d", sum);
    return 0;
}
```

Output

Enter any number : 432
Sum of digits : 9

Q) Write a program that read an Integer number and reverse it.

theory

⇒

$$num = 123$$

$$num = 0$$

$$temp = num = 123 \quad q = num/10$$

$$\text{while}(\text{temp} \neq 0) \quad mnum = q * 10 + r$$

$$\{ \quad r = temp \% 10$$

$$num = num * 10 + r \quad mnum = mnum + r$$

$$temp = temp / 10 \quad q = mnum/10$$

$$\} \quad \}$$

$$\begin{array}{r} 10 \\ | \\ 123 \\ | \\ 10 \\ \hline 23 \\ | \\ 20 \\ \hline 3 \end{array}$$

$$\begin{array}{r} 20 \\ | \\ 12 \\ | \\ 10 \\ \hline 2 \\ | \\ 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 20 \\ | \\ 1 \\ | \\ 0 \\ \hline 1 \end{array}$$

coding

⇒ #include <stdio.h>

int main()

{ int num, temp, r; num=0;

printf("Enter any number : ");

scanf("%d", &num);

temp = num;

while(temp != 0)

{ r = temp % 10;

num = num * 10 + r;

temp = temp / 10;

}

printf("Reverse number = %d", num);

return 0;

}

output

Enter any number : 543
Reverse number = 345

* **Palindrome number:** A palindrome number is a number that reads the same backward as forward. In other words, it remains unchanged when its digits are reversed. For example 121, 3443 etc.

* Write a C program that read a integer number and check if it's a palindrome number or not.

Coding

⇒ `#include <stdio.h>`

int main()

{ int num, temp, n, num=0, reverse;
printf ("Enter any number : ");
scanf ("%d", &num);

temp = num;

while (temp != 0)

 n = temp % 10;

 num = num * 10 + n;

 temp = temp / 10;

}

reverse = num;

if (reverse == num)

 printf ("Palindrome number\n");

else

 printf ("Not palindrome number\n");

return 0;

}

Output

⇒ Enter any number : 313
Palindrome number.

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

that's work

④ Armstrong Number: An n-digit number is an Armstrong number if the sum of its individual digits, each raised to the power of n, is equal to the number itself.

for example:

$$153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$$

$$4356 = 4^4 + 3^4 + 5^4 + 6^4 = 256 + 81 + 625 + 1296 = 2256$$

check

④ Write a program that read any number and check if it's armstrong number or not.

Coding

```
#include <stdio.h>
int main()
{
    int num, temp, n, sum=0;
    printf("Enter any number : ");
    scanf("%d", &num);
    temp = num;
    while (temp != 0)
    {
        n = temp % 10;
        sum = sum + n * n * n;
        temp = temp / 10;
    }
    if (sum == num)
        printf("Armstrong number");
    else
        printf("Not armstrong number");
}
```

Output

⇒ Enter any number: 37037
Armstrong number.

* Write a program that read initial number and final number and check armstrong number into initial number to final number.

```
#include <ntdio.h>
int main()
{
    int = initialNum, finalNum, i, temp, num=0;
    printf("Enter initial Number: ");
    scanf("%d", &initialNum);
    printf("Enter final Number: ");
    scanf("%d", &finalNum);
    for(i=initialNum; i<=finalNum; i++)
    {
        temp=i;
        while(temp!=0)
        {
            r=temp%10;
            num=num+r*r*r;
            temp=temp/10;
        }
        if(num==i)
            printf("Armstrong Number: %d", num);
    }
    return 0;
}
```

Output:

Enter initial Number: 1
Enter final Number: 1000
Armstrong Number: 1
Armstrong Number: 153
Armstrong Number: 370
Armstrong Number: 371
Armstrong Number: 407

Q Write a program that read any number, and display digit of this number.

$\Rightarrow \#include <stdio.h>$

int main()

{ int num, n, count = 0;
printf("Enter any number : ");

n = num; scanf("%d", &num);
while(n != 0)

{ n = n / 10;
count++;

printf("The digit of this number : %d", count);

return 0;

3
Output: Enter any number : 1253
The digit of this number : 4

Q Write a C program that read any number, and determine strong number.

defining strong number:

$\Rightarrow \#include <stdio.h>$

int main()

{ int num, n, i, temp, num=0, fact;

printf("Enter any number : ");

scanf("%d", &num);

while(temp == num);

while(temp != 0)

{ n = temp % 10;

fact = 1;

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

{ fact = i * fact;

}

145 strong number

= 145

= 1 + 4 + 5

= 1! + 4! + 5!

= 1 + 24 + 120

= 145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

145

```

num = num + fact;
temp = temp / 20;
if (num == num)
    printf (" %d is a strong number.\n", num);
else
    printf (" %d is not a strong number.\n", num);

```

3-

Output:

Enter any number: 145

145 is a strong number.

~~④ Write a c program~~

~~Serien~~

~~④ Write a c program that read for given below~~

~~derien.~~

~~1 + 2 + 3 + ... + n~~

~~minimum task soft work~~

~~it:~~

~~minimum soft to remove task soft work~~

~~(min * "bx + ... + f + e")~~

~~(min >= 0)~~

~~l + min = min~~

~~l + d = d~~

Seriens

④ Write a c program to find the sum of the following seriens. (using for loop)

$$1+2+3+\dots+n$$

⇒ `#include <stdio.h>`

`int main()`

{ int num, sum=0;

printf("Enter the last number of the seriens : ");

scanf("%d", &num);

printf("1+2+3+\dots+%d", num);

for(i=1; i<=num; i=i+1)

{ num = num + i;

printf("%d", num);

getch();

Output: Enter the last number of the seriens : 10
3. 1+2+3+\dots+10 = 55.

④ Write a c program to find the sum of the following seriens. (using while loop)

$$3+5+7+\dots+n$$

⇒ `#include <stdio.h>`

`int main()`

{ int num, a=3, sum=0;

printf("Enter the last number of the seriens : ");

scanf("%d", &num);

printf("3+5+7+\dots+%d", num);

while(a<=num)

{ num = num + a;

 a=a+2;

}

```
printf("%d\n", num);  
getch();
```

3
Output: Enter the last number of the series : 20

$$3+5+7+\dots+20 = 99.$$

④ Write a C program to find the value of the following series.

$$3 \cdot 4 + 5 \cdot 6 + 7 \cdot 8 + \dots + n_1 \cdot n_2$$

```
#include <stdio.h>  
int main()
```

```
{ int n1, n2, a=3, b=4, num=0;
```

~~printf("Enter the last number of the series : ");~~

~~scanf("%d %d", &n1, &n2);~~

```
printf("3 \cdot 4 + 5 \cdot 6 + 7 \cdot 8 + \dots + %d \cdot %d ", n1, n2);
```

~~while (a <= n1 && b <= n2)~~

```
{ num = num + a * b;
```

~~a = a + 2;~~

~~b = b + 2;~~

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

~~return 0;~~

3

Output: Enter n1 and n2 : 10, 11

$$3 \cdot 4 + 5 \cdot 6 + 7 \cdot 8 + \dots + 10 \cdot 11 = 188$$

④ Write a C program to find the value of $1^2 + 2^2 + 3^2 + \dots + N^2$.

⇒ #include <stdio.h>
int main()
{
 float i, n, num = 0;
 printf("Enter floating number of n: ");
 scanf("%f", &n);
 printf("1^2 + 2^2 + 3^2 + ... + %d^2 = ", n);
 for(i = 1; i <= n; i = i + 1)
 {
 num = num + i;
 }
 printf(" = %.1f", num);
}

Output: Enter floating number of n: 10.5
 $1^2 + 2^2 + 3^2 + \dots + 10.5^2 = 602.0$

④ Write a C program to find the value of $1^2 + 2^2 + 3^2 + \dots + N^2$.

⇒ #include <stdio.h>
int main()
{
 int n, num = 0, i;
 printf("Enter n: ");
 scanf("%d", &n);
 for(i = 1; i <= n; i = i + 1)
 {
 num = num + i * i;
 }
 printf("1^2 + 2^2 + 3^2 + ... + %d^2 = ", n);
 printf(" = %.d/n", num);
 return 0;
}

Output: Enter n = 10
 $1^2 + 2^2 + 3^2 + \dots + 10^2 = 385$

* Write a C program to find the value of this series.

$$1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

$\Rightarrow \#include <stdio.h>$
int main()

```
{ double i, n, num=0;
printf("Enter value of the n : ");
scanf("%d", &n);
printf("1 + 1/2 + 1/3 + ..... + 1/n = ", n);
for(i=1; i<=n; i++)
{
    num = num + 1/i;
}
printf("= %.2f", num);
```

3
Output: Enter ~~of~~ the value of the n :
 $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{6} = 2.45$

* Write a C program to find the value of this series.

$$1 * 2 * 3 * \dots * n$$

$\Rightarrow \#include <stdio.h>$

```
int main()
{
    int n, i, result=1;
    printf("Enter value of n : ");
    scanf("%d", &n);
    printf("1 * 2 * 3 * ..... * %d = ", n);
    for(i=1; i<=n; i++)
    {
        result = result * i;
    }
    printf("= %d", result);
```

3
Output: Enter value of n : 6
 $1 * 2 * 3 * \dots * 6 = 720$

④ Write a C program to find the value of this pattern

1 2 3 ... n

#include <stdio.h>

int main

{ int n, i, result = 1

printf("Enter the value of n : ");

scanf("%d", &n);

printf("1x2x3x...x%d", n);

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

{ result = result * i; }

result = result * i;

printf("%d", result);

Output: Enter the value of n

1x2x3x...

④ Write a C program to find the value of this pattern

1-2+3-4+5-6...

#include <stdio.h>

int main()

{ int n, i, even=0, odd=0

printf("Enter the value of n : ");

if (n%2 == 0)

printf("1-2+3-4+5-6... (%d)", n);

else

printf("1-2+3-4+5-6... (%d)", n);

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

{ if (i%2 == 0)

even = even + i;

else
 odd = odd + i;

odd = 0; start : fibo
odd = 1; minig increment

④ Fibonacci

④ Fibonacci numbers: The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding ones, typically starting with 0 and 1: An like:

0, 1, 1, 2, 3, 5, 8, 13, 21

$f_0 = 0$
 $f_1 = 1$

~~first~~ → first = 0
second = 1

→ Fibonacci = first + second

first = second

second = Fibonacci

④ Write algorithm, C program and flow chart to find nth number of Fibonacci series.

Algorithm:

Step-1: start

Step-2: set first = 0
 second = 1
 count = 0

Step-3: input n

Step-4: while count < n repeat

 Step-5, 6, 7

 Step-5: if count <= 1

 i) Yes, fibo = count

 ii) No, fibo = first + second

 first = second

 Step-6: print fibo second = fibo

 Step-7: count++; loop-8: End

C program

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

```
int main()
```

```
{ int first = 0, second = 1, count = 0,
```

 fibo, n;

```
printf("Enter range : ");
```

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

```
while (count < n) → printf("Fibonacci
```

```
    number : ");
```

```
{ if (count <= 1)
```

 fibo = count;

 else {

 fibo = first + second;

 first = second;

 second = fibo;

```
        printf("%d", fibo);
```

```
        count++;
```

}

}

Lucan Neriens: The Lucan series is similar to the Fibonacci sequence but starts with different initial numbers. Instead of starting with 0 and 1 like the Fibonacci sequence, the Lucan sequence starts with 2 and 1. An like:

2, 1, 3, 4, 7, 11, 18, 29

$$\text{first} = 2$$

$$\text{second} = 1$$

$$= \text{Lucan} = \text{first} + \text{second}$$

$$\text{first} = \text{second}$$

$$\text{second} = \text{Lucan}$$

★ Write a C program to find n^{th} term of Lucan series.

⇒ #include <stdio.h>

int main()

{ int first = 2, second = 1, count = 0; } $\text{Lucan}, n;$

printf("Enter the value of n:"); $\text{scanf}(\text{"%d"}, \&n);$

printf("Lucan series = 2, 1, ");

for (i = 3; i <= n; i++)

{ $\text{Lucan} = \text{first} + \text{second};$

 first = second;

 second = Lucan; }

printf("%d", Lucan);

3
S
B

Output: Enter the value of n : 8

Lucan series : 2, 1, 3, 4, 7, 11, 18, 29

Pattern (1-7) \rightarrow Right angle triangle.

Pattern type - 1

① Pattern

```
#include <stdio.h>
int main()
{
    int n, row, col;
    printf("Enter number of row: ");
    scanf("%d", &n);
    for (row = 1; row <= n; row++)
    {
        for (col = 1; col <= row; col++)
            printf("%d", col);
    }
}
```

3 3
3

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

3
1
1 2
1 2 3

Output: Enter number of row: 3

Pattern

② #include <stdio.h>

```
int main()
```

```
{ int n, row, col;
```

```
printf("Enter value of row: ");
```

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

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

```
{ for (col = 1; col <= row; col++)
```

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

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

3 3

Output:

Enter value of row: 4

1 2 3 4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

1

2

3

4

Pattern-3

⇒ `#include <ntdio.h>`
 int main()

```
{ int n, row, col;
printf("Enter value of row: ");
scanf("%d", &n);
for( row=1; row<=n; row++)
{ for( col=1; col<=row; col++)
{ if (col%2==0)
  printf("0");
else
  printf("1");
}
printf("\n");
}
```

Output:

Enter value of row: 4
 1 0 1 0
 1 0 1 0
 1 0 1 0
 1 0 1 0

Pattern-4

⇒ Code:

`#include <ntdio.h>`
 int main()

```
{ int n, row, col;
printf("Enter value of row: ");
scanf("%d", &n);
for( row=1; row<=n; row++)
{ for( col=1; col<=n; col++)
{ if (row%2==0)
  printf("0");
else
  printf("1");
}
printf("\n");
}
```

Output:

Enter value of row: 5
 1 0 1 0 1
 0 1 0 1 0
 1 0 1 0 1
 0 1 0 1 0
 1 0 1 0 1

Pattern - 5

⇒ Coding:

```
#include <stdio.h>
int main()
{ int n, row, col;
printf("Enter the value of row:");
scanf("%d", &n);
for( row=1; row<=n; row++)
{ for( col=1; col<=row; col++)
{ printf("%c", col+64);
}
printf("\n");
}
}
```

Output:

Enter the value of row
A
A B
A B C
A B C D

Pattern - 6

⇒ Coding:

```
#include <stdio.h>
int main()
```

```
{ int n, row, col;
printf("Enter the value of row:");
scanf("%d", &n);
```

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

```
{ for( col=1; col<=row; col++)
{
```

```
printf("%c ", (row+64));
}
```

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

Output:

Enter the value of row

A

B B

C C C

D D D

E E E E E

Pattern - 7

⇒ Coding:

```
#include <ntdio.h>
int main()
{ int n, row, col;
printf("Enter the value of row : ");
scanf("%d", &n);
for (row=1; row<=n; row++)
{ for (col=1; col<=row; col++)
{ printf("*");
    printf("\n");
}
}
}
```

Output:

Enter the value of row = 3

*
* *
* * *

Pattern: 8

[Pattern type - (2)]

⇒ Coding:

```
#include <ntdio.h>
int main()
{ int n, row, col;
printf("Enter the value of row : ");
scanf("%d", &n);
for (row=n; row>=1; row--)
{ for (col=1; col<=row; col++)
{ printf("%d", col);
    printf("\n");
}
}
}
```

Output:

Enter the value of row = 5

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

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

Pattern (8-12) \rightarrow Left angle triangle

Pattern - (9)

\Rightarrow coding:

```
#include <stdio.h>
int main()
{
    int n, row, col;
    printf("Enter the value of row : ");
    scanf("%d", &n);
    for (row = n; row >= 1; row--)
    {
        for (col = 1; col = row; col++)
        {
            printf("%d ", row);
            printf("\n");
        }
    }
}
```

3

Output:

Enter the value of row:

4 4 4 4 4

3 3 3 3 3

2 2 2 2 2

1

Pattern - (10)

\Rightarrow coding:

```
#include <stdio.h>
int main()
{
    int n, row, col;
    printf("Enter the value of row : ");
    scanf("%d", &n);
    for (row = n; row >= 1; row--)
    {
        for (col = 1; col <= row; col++)
        {
            if (col % 2 == 0)
                printf("0 ");
            else
                printf("1 ");
        }
        printf("\n");
    }
}
```

3

Output:

Enter the value of row : 5

1 0 1 0 1

1 0 1 0

1 0 1

1 0

1

Pattern: 1

⇒ coding:

```
#include <stdio.h>
int main()
```

```
{ int n, row, col;
```

printf("Enter the value of row : ");

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

```
for (row = n; row >= 1; row--)
```

```
{ for (col = 1; col <= row; col++)
```

```
{ printf("%c", c4 + col);
```

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

```
}
```

Output:

Enter the value of row: 4

A B C D,

A B C

A B

A

Pattern-1/2

⇒ coding:

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

```
int main()
```

```
{ int n, row, col;
```

printf("Enter the value of row : ");

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

```
for (row = n; row >= 1; row--)
```

```
{ for (col = 1; col <= row; col++)
```

```
{ printf("%c", c4 + row);
```

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

```
}
```

```
}
```

Output:

Enter the value of row = 5

E E E E E

D D D D

C C C

B B

A

Pattern type - (3)

Pattern-13

⇒ coding

```
#include <ntdio.h>
int main ()
```

```
{ int n, row, col;
```

```
printf ("Enter the value of n : ");
```

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

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

```
{ for (col = 1; col <= row; col++)
```

```
{ printf ("%d ", col);
```

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

```
3 for (row = n-1; row >= 1; row--)
```

```
{ for (col = 1; col <= row; col++)
```

```
{ printf ("%d ", col);
```

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

Output: Enter the value of n : 5

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

Pattern: 14

⇒ coding:

```
#include <stdio.h>
int main()
```

```
{ int n, row, col;
printf("Enter the value of n: ");
scanf("%d", &n);
for(row=1; row<=n; row++)
{ for(col=1; col<=row; col++)
{ printf("%d", row);
    if(col==row)
        printf("\n");
}
}
```

Output: Enter the value of n: 4

1					A
2	2				A
3	3	3			A
4	4	4	4	D	A
3	3	3	3	D	A
2	2			D	A
1				D	A

Pattern - (15)

→ Coding:

```
#include<stdio.h>
int main()
{
    int n, row, col;
    printf("Enter the value of n: ");
    scanf("%d", &n);
    for(row=1; row<=n; row++)
    {
        for(col=1; col<=row; col++)
        {
            printf("%c", 'A'+col);
        }
        printf("\n");
    }
    for(row=n; row>=1; row--)
    {
        for(col=1; col<=row; col++)
        {
            printf("%c", 'A'+col);
        }
        printf("\n");
    }
}
```

Output:

N:4
Enter the value of n: 4

A

A B

A B C

A B C D

A B C

A B

A

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
A B C D E
A B C D E

Pattern - (16)

Pattern type - (4)

⇒ Coding:

```
#include <stdio.h>
int main()
```

```
{ int n, now, col;
```

printf("Enter the value of n: ");

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

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

```
{ for(col=1; col<=n-now; col++)
```

```
{ printf(" . "); } [Center if (.) ]
```

```
for(col=1; col<now; col++)
```

```
{ printf("%d ", col); }
```

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

```
}
```

```
}
```

Output:

Enter the value of n: 5

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

Pattern - (27)

→ coding:

```
#include <stdio.h>
int main()
```

```
{ int n, row, col;
```

printf("Enter the value of n: ");

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

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

```
    for (col = 1; col <= n - row; col++)
```

```
        if (row % 2 == 0)
```

```
            printf("0 ");
```

```
        else
```

```
            printf("1 ");
```

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

```
}
```

Output:

Enter the value of n: 4

```
1 0 0 0
  1 1 1 1
    0 0 0 0
      1 1 1 1
```

Pattern 27

Pattern - (10)

→ coding:

```
#include <ntdio.h>
int main()
```

```
{ int n, row, col;
printf("Enter the value of n:");
scanf("%d", &n);
for (row=1; row<=n; row++)
{ for (col=1; col<=n-row; col++)
    { printf(" ");
    for (col=1; col<=row; col++)
        { printf("%c", col+'A');
    printf("\n");
}
}
}
```

output: Enter the value of n: 4

```

      A   S   D
      A   B   S
      A   B   C
      A   B   C   D
```

Pattern - (1)

Pattern type - (5)

⇒ coding

```
#include <stdio.h>
int main()
```

```
{ int n, row, col;
```

printf(" Enter the value of n : ");

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

```
for (row = n; row >= 1; row--)
```

```
{ for (col = 1; col <= n - row; col++)
```

```
{ printf(" . . ");
```

```
3 for (col = 1; col <= row; col++)
```

```
{ printf("%d", col);
```

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

```
}
```

Output: Enter the value of n : 4

1 2 3 4

1 2 3

4 A

3 A

2 A

1 A

B A

(C) padding(1)-(2)

→ coding:

```
#include <stdio.h>
int main()
{
    int n, row, col;
    printf("Enter the value of n : ");
    scanf("%d", &n);
    for(row = n; row >= 1; row--)
        for(col = 1; col <= n - row; col++)
            printf(".");
    for(col = 1; col <= row; col++)
    {
        if(col % 2 == 0)
            printf("0");
        else
            printf("1");
    }
    printf("\n");
}
```

3
Output: Enter the value of n : 4

```
1 0 1 0
1 0 1 1 1 1
0 1 0 0
0 0 1
0
A
```

1, 2, 3, 4

Pattern-(2)

⇒ coding:

```
#include <stdio.h>
int main()
{ int n, row, col; // n to sub n : ;
    printf("Enter the value of n : ");
    scanf("%d", &n);
    for (row = n; row >= 1; row--) // n = won
    {
        for (col = 1; col <= n - row; col++) // string
        {
            printf("%c", row + col); // string
        }
        printf("\n"); // string
    }
}
```

Output: Enter the value of n : 5

```

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

Pattern - (23) Pattern type - (6)

→ coding:

```
#include <stdio.h>
int main()
{
    int n, row, col;
    printf("Enter the value of n : ");
    scanf("%d", &n);
    for (row = 1; row <= n; row++)
    {
        for (col = 1; col <= n - row; col++)
            printf(".");
        for (col = 2; col <= n - row; col++)
            printf("%d", col);
        printf("\n");
    }
}
```

Ex: or to solve
A 3
B 3
C 0 0
D 8 8
E A

Output!

Enter the value of n : 4

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

Pattern - (23)

⇒ coding:

```
#include <stdio.h>
int main()
```

```
{ int n, row, col;
printf("Enter the value of n: ");
scanf("%d", &n);
for (row = 1; row <= n; row++)
{ for (col = 1; col <= n - row; col++)
    { printf(" . ");
    }
    for (col = 2; col <= row; col++)
    { printf("%c", row + col);
    }
    printf("\n");
}
for (row = n - 1; row >= 1; row--)
{ for (col = 1; col <= n - row; col++)
    { printf(" . ");
    }
    for (col = 2; col <= row; col++)
    { printf("%c", row + col);
    }
    printf("\n");
}
```

Output:
Enter the value of n: 3

A
B B
C C C
B B
A

To solve, start
3
2 2
1 1 1
0 0 0 0
For n

Pattern - (24) Pattern type - (7)

Coding:

```
#include <ntdio.h>
int main()
{ int n, row, col;
printf("Enter the value of n: ");
scanf("%d", &n);
for (row = 1; row <= n; row++)
{ for (col = 1; col <= n; col++)
{ printf("%d ", col);
}
printf("\n");
}
}
```

Output:

Enter the value of n: 4

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

Pattern - (25)

Coding:

```
#include <ntdio.h>
int main()
{ int n, row, col;
printf("Enter the value of n: ");
scanf("%d", &n);
for (row = 1; row <= n; row++)
{ for (col = 1; col <= n; col++)
{ printf("%c ", row + col);
}
printf("\n");
}
}
```

Output:

Enter the value of n: 5

A	A	A	A	A
B	B	B	B	B
A	A	B	B	C
B	C	A	B	B
C	B	B	A	B

Pattern - (26)

⇒ coding:

```
#include <stdio.h>
int main()
{ int n, row, col;
  printf("Enter the value of n : ");
  scanf("%d", &n);
  for(row = 1; row <= n; row++)
  { for(col = 1; col <= row; col++)
    { printf("%d ", row * col); }
  printf("\n"); }
}
```

(25) - Pattern

Output:

Enter the value of n :

1
2 4

3 6 9
4 8 12 16

5 10 15 20 25

6 12 18 24 30 36

7 14 21 28 35 42

8 16 24 32 40 48

Pattern - (27)

⇒ Coding:

```
#include <stdio.h>
int main()
{
    int n, row, col;
    printf("Enter the value of n : ");
    scanf("%d", &n);
    for(row=1; row<=n; row++)
    {
        for(col=1; col<=n-row; col++)
        {
            printf(" . ");
        }
        for(col=1; col<=2*row-1; col++)
        {
            printf("%d ", col);
        }
        printf("\n");
    }
}
```

Output

Enter the value of n : 4

```

      1
    1 2 3
  1 2 3 4 5
1 2 3 4 5 6 7
```