

History of C :-

The C programming language is a structure Oriented programming language, developed at Bell laboratories in 1972 by "Dennis Ritchie".

C programming language features were derived from an earlier language called "B" (Basic Combined Programming language - BCPL)

Structure of C :-

- Pre Processor directive.
- Variables (Global)
- User defined functions
- main()
 - {
 - variables (Local)
 - Statements;
 - Expressions;
 - }

Pre Processor directive :-

```
#include <stdio.h>
#include <conio.h>
Void main()
{
    clrscr();
    printf("Helloworld");
    getch();
}
```

→ linking

include <stdio.h>



holders

Global Variables :-

Def :- The variables which are declared outside any function is known as a Global Variable.

default Value :- zero(0)

Scope :- Throughout the program (anywhere).

Ex :-

```
#include<stdio.h>
#include<conio.h>
int a,b,c;
void main()
{
    clrscr();
    printf("enter the value of a & b");
    scanf("%d %d", &a, &b);
    c = a+b;
    printf("%d + %d = %d", a, b, c);
    getch();
}
```

Local Variable :-

Def :- The variables which are declared inside any function is known as a Local Variable.

default Value :- Garbage Value

Scope :- Within the block.

Ex :- (i)

```
#include<stdio.h>
#include<conio.h>
void main()
{
```

```

        int a;
        clrscr();
        printf("%d", a); // Garbage value
    } getch();
}

```

(ii) #include <stdio.h>

#include <conio.h>

int a=20;

void main()

{

int a=10;

clrscr();

printf("%d", a); // a=10,

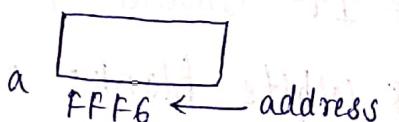
getch();

}

printf("%d", a); // a=20;

Variables :-

Def :- Variables are the "identifiers" which represents the 'address' of 'Memory locations'.



Identifiers :- Unique name given to a particular block/ used in 'C'.

- * 8 bits - 1 byte

- * 1024 bytes - 1 KB

- * 1024 KB - 1 MB

- * 1024 MB - 1 GB

- * 1024 GB - 1 TB.

Keywords :-

"Keywords" are pre-defined reserved words used in Programming.

- Compiler known words
- Keywords are part of the Syntax and they cannot be used as an "identifier".

Ex:- int, char, float, for, if, while, include, auto, break, register

In C88 → 32 keywords

C98 → 32 + 5 ⇒ 37 keywords

C11 → 32 + 5 + 7 → 44 keywords

Rules :-

- * Keywords are not allowed.
- * Alphabets and numbers are allowed.
- * Special characters are not allowed except '-' (Underscore).
- * It should not start with number
- * White blanks or Spaces are not allowed.
- * Length is 32.

ANSI :-

ANSI → American Nation Standards Institute, in 1918.

ASCII values:-

Uppercase 'A' has ASCII value '65' in decimal, So for 'Z', the value is '90' in decimal.

Lowercase 'a' has ASCII value '97' in decimal, So for 'z', the value is '122' in decimal.

→ for (values) numbers 0 - 48, 1 - 49, 2 - 50, ... 9 - 57.

and for 'A' → 65 ; 'B' → 66 ; 'C' → 67 etc.

Datatype :-

Def:- Datatypes are defined as the data storage format that a variable can store a data to perform a specific operation.

Datatypes are classified into two types.

- * Primary datatype
- * Secondary datatype

Primary datatype :-

In Primary there are 5 basic datatypes. They are

* int, char, float, double and void.

The storage size and range for int and float datatype will vary depend on the CPU processor (8, 16, 32 and 64 bit).

C datatypes/storage size	Range
char/1	-127 to 127
int/2	-32,767 to 32,767
float/4	1E-37 to 1E+27 with 16 digits of precision
double/8	1E-37 to 1E+27 with 10 digits of precision
long double/10	1E-37 to 1E+27 with 10 digits of precision
long int/4	-2,147,483,647 to 2,147,483,647
short int/2	-32,767 to 32,767
unsigned short int/2	0 to 65,535
Signed short int/2	-32,767 to 32,767
void	no size.

Integer datatype:-

It allows a variable to store numeric values.

→ "int" keyword is used to refer integer datatype

→ Storage size of int datatype is 2 or 4 or 8 byte

→ if we use 16 bit processor, 2 byte (16 bit) memory will be allocated for int data type,

Similarly, 4 byte of memory for 32 bit processor

8 byte of memory for 64bit Processor

Character datatype :-

It allows a variable to store only one character.

→ "char" keyword is used to refer character datatype

→ Storage size of char datatype is '1' byte

Floating datatype:-

Floating point datatype consists of 2 types. They are :-

1. float

2. double

float:-

It allows a variable to store decimal values.

→ "float" keyword is used to refer float datatype.

→ Storage size of float datatype is 4 byte

→ We can use upto 6 digits after decimal using float datatype

Ex:- 10.456789.

double :-

"double" is same as float datatype which allows upto 10 digits after decimal.

Void datatype :-

'Void' is an Empty datatype that has no value. This can be used in 'functions' and 'pointers'.

(ii) Secondary datatype :-

In Secondary datatype there are 5+1 types. They are

- * Arrays * Pointers
- * Structures
- * Unions
- * Enumerations
- * Array, Pointer, Structure and Union are called derived datatype in 'C' language.

int Size and format:-

Size/range → 2^{15} (-32768 to +32767)

format:-

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

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

```
main()
```

```
{     int a,b,c ; → declaration
```

```
    a=10; } → initialisation
```

```
    b=20;
```

```
    c=a+b; → Utilisation
```

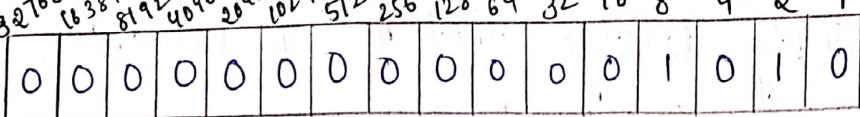
```
    printf("%d", c);
```

$a = 20;$ } → re-initialization

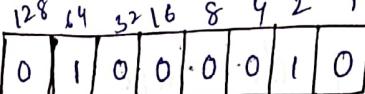
$b = 50;$ }

int a=10

Ex:- int a=10;

→  → 10B
 msb | 0 0 0 0 0 0 0 0 0 0 1 0 1 0 |
 10

Char Range :- -128 to +127

→  → B=65
 128 64 32 16 8 4 2 1
 0 1 0 0 0 0 1 0

format:-

%d → int ; %i → Everything

%c → char ; %lf → double

%s → string ; %Lf → Long double

%f → float ; %p → address (pointer)

%x → Hexadecimal ; %u → Unsigned ;

%e → exponential.

(ii) #include <stdio.h>

#include <conio.h>

void main()

{

char ch='A';

printf("%c", ch); // A

printf("%d", ch); // 65

}

int a=1025;

printf("%d", a); // 1025

printf("%c", a); // ☺

Comparision between '<, >' and "

file1.c

#include <stdio.h>

#include <conio.h>

int a=20;

main()

{ -- -- -- }

file2.c

#include <stdio.h>

#include <conio.h>

#include "file1.c"

main()

{ extern int a;

printf("%d", a); // 20

Program

1. Write a 'c' program to perform all arithmetic Operations.

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int n1, n2, sum, Sub, mult, mod;
    float div;
    printf("Enter any two numbers\n");
    Scanf("%d%d", &n1, &n2);
    Sum = n1 + n2;
    Sub = n1 - n2;
    mult = n1 * n2;
    mod = n1 % n2;
    div = n1 / n2;
    printf("Sum = %d\n", Sum);
    printf("diff = %d\n", Sub);
    printf("Prod = %d\n", mult);
    printf("quotient = %f\n", div);
    printf("modulus = %d", mod);
    return 0;
}
```

Operators

Operator :-

Def :- An Operator is a symbol that tells the Compiler to perform a certain mathematical or logical Operations.

Operators are classified into 2 types :-

1. Based on number of Operands

2. Based on Operation.

1. Based on number of Operands :-

(i) Unary → One Operand

(ii) binary → 2 Operands

(iii) Ternary → 3 operands

(i) Unary Operator :-

(a) Post :- first use the value then update the value.

(b) Pre :- first update the value then use the value.

(a) Post :- 1. Post Increment ($a++$)

2. Post decrement ($a--$)

Ex :- if $a=5$

$a++$ then $a=6$

$a--$ then $a=4$

(b) Pre :- 1. Pre Increment ($++a$)

2. Pre decrement ($--a$)

Ex :- if $a=5$

$++a$ then $a=6$

$--a$ then $a=4$

(i) #include <stdio.h>
 #include <conio.h>
 void main()
 {
 int a=6, b=7;
 a++; } independent lines
 b++; }
 --a; }
 --b; }
 b--; }
 printf("%d %d", a, b); // a=6, b=6
 }

(ii) #include <stdio.h>
 #include <conio.h>
 void main() {
 int a=6, b=7;
 a = a++ + b++; // 14, 8 } dependent lines
 b = b++ + a++; // 15, 23 }
 printf("%d %d", a, b);
 }

Ex:- a=3, b=4, c=5

$$a = a++ + ++b + c++$$

$$b = c++ + a++ - b++$$

$$c = b++ + f + a + c++$$

$$\therefore a = 16, b = 17, c = 40$$

a 18 14 15 16

b 4 8 18 16 17

c 5, 6, 7 39 40

Ternary Operator :-

Syntax :-

```

  (Condition) ? (Statement 1) : (Statement 2);
```

True
False

Ex:- $(a > b) ? \text{printf}(“a is greater”): \text{printf}(“b is greater”);$

(2) $(a > b \& a > c) ? \text{printf}(“a is greater”): ((b > c) ? \text{printf}(“b is greater”): \text{printf}(“c is greater”));$

Nested Ternary Operator :-

$(a > b) ? ((a > c) ? \text{printf}(“a is greater”): \text{printf}(“c is greater”)): ((b > c) ? \text{printf}(“b is greater”): \text{printf}(“c is greater”));$

2. Based on Operation :-

(i) Arithmetic Operator :-

→ +, -, *, /, %

(ii) Relational Operator :-

→ >, <, >=, <=, ==, !=

(iii) Logical Operator :-

→ &&, ||, !

0	0	0
0	1	0
1	0	0
1	1	1

$\&\&$ (logical and)

0	0	0
0	1	1
1	0	1
1	1	1

|| (logical or)

0	1
1	0

! (logical not)

(iv) Bitwise Operator :- &, |, ~, ^, <<, >>.

(a) bitwise & : write a program to compare two numbers

Ex:- $a = 4, b = 6$ then

a & b is 4 - 0100

$$\begin{array}{r} 6 - 0110 \\ \hline 0100 \rightarrow 4 \end{array}$$

(b) bitwise (or) / :-

Ex :- $a = 4, b = 6$ then

alb is 4 - 0100

$$\begin{array}{r} 6 \\ - 0110 \\ \hline 0110 \end{array} \rightarrow 6$$

(c) bitwise not (\sim) :- $\eta = 4 - 0100$

$$\text{~}\approx\eta = 1011 - \text{i's comp}$$

(d) bitwise x-or(1) :- Ex :- a=10, b=8

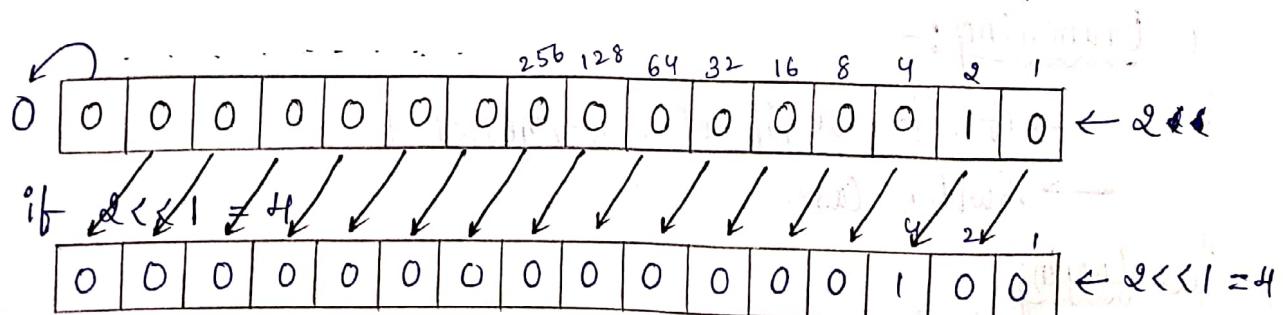
0	0	0
0	1	1
1	0	1
1	1	0

$$a - 1010 - 10$$

$$\begin{array}{r} 1 \\ b - \underline{1000} - 8 \\ 0010 - 2 \end{array}$$

(e) Bitwise leftshift ($<<$) :-

number << no. of bits to be shifted.

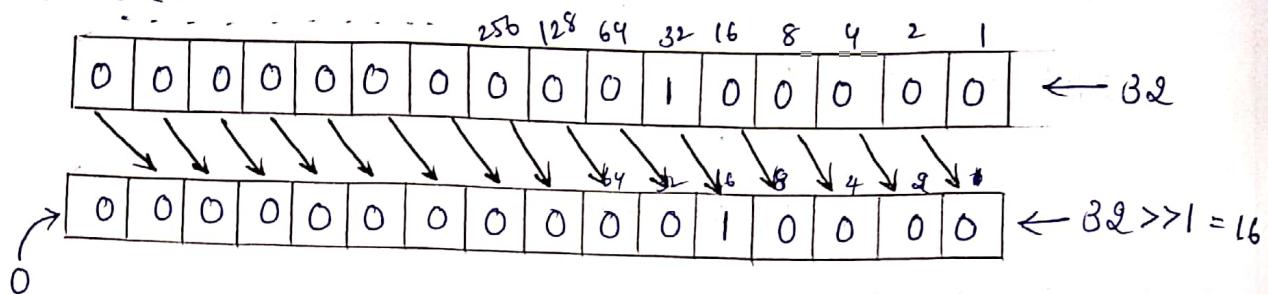


Formula $\rightarrow a \cdot x^b$

(f) Bitwise Rightshift ($>>$):-

number $>>$ no. of bits to be shifted.

if $32 >> 1$



formula:- $a/2^b$ where b - no. of bits.

Control

(v) Assignment Operator :-

$\rightarrow =, +=, *=, -=$

Ex:- $a+1$; $a-1$; $a \wedge b$
 $a=a+1$; $a=a-1$; $a=a \wedge b$.

Control Statements :-

Def:- A control statement is a statement that determines whether other statements will be executed.

\rightarrow CS which is used to control the flow of program.

Two types of Control statements are:-

1. Branching

2. Looping.

1. Branching:-

\rightarrow if, if else, if else if, nested if.

\rightarrow Switch Case

2. Looping:-

\rightarrow for loop

\rightarrow while loop

\rightarrow do-while loop

1. Branching :-

(a) if :-

Syntax :-

```
| if (Condition)
| {
|   //statements //
| }
```

(b) if else :-

Syntax :-

```
| if (Condition)
| {
|   //statements //
| }
| else
| {
|   //statements //
| }
```

(c) if else if :-

Syntax :-

```
| if (Condition)
| {
|   //statements //
| }
| else if (Condition)
| {
|   //statements //
| }
| else
| {
|   //statements //
| }
```

(d) Nested if

Syntax :-

```
if (condition)
{
    if (condition)
    {
        //statements//
    }
}
```

(ii) Switch Case :-

Syntax :-

```
switch(n)
{
    Case 1 : //statements//
        break;
    Case 2 : //statements//
        break;
    .
    .
    .
    Case n : //statements//
        break;
    default : //statements//
}
```

(ii) Syntax :-

```
Switch(n)
{
    default: //statements//
        break;
    Case 1 : //statements//
        break;
    Case 2 : //statements//
        break;
    .
    .
    .
    Case n : //statements//
}
```

Q. Looping :-

(a) for loop :-

Syntax :-

```
for (initialization; condition; updation)
{
    //statements//
}
```

(b) while loop :-

Syntax :-

```
While (condition)
{
    //statements//
}
```

(c) do-while loop :-

Syntax :-

```
do {  
    // statements  
} while (condition);
```

Programs :-

1. Write a program to add 2 numbers without using '+' symbol.

```
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
    clrscr();  
    int a, b;  
    printf("enter two numbers\n");  
    Scanf("%d %d", &a, &b);  
    Sum = a - (-b);  
    printf("%d", Sum);  
    getch();  
}
```

2. WAP to Subtract 2 numbers without using '-'.

```
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
    clrscr();  
    int a, b;  
    printf("enter two values\n");
```

```

    Scanf("%d %d", &a, &b);
    c = a + (b + 1);
    printf("%d", c);
    getch();
}

```

3. WAP to check whether the provided 2 integers are equal.

```

#include <stdio.h>
#include <conio.h>
void main()
{
    clrscr();
    int a, b;
    printf("enter two values\n");
    Scanf("%d %d", &a, &b);
    if (a == b)
    {
        printf("a and b are equal");
    }
    else
    {
        printf("a and b are not equal");
    }
    getch();
}

```

4. WAP to compare 2 numbers and to print largest among those numbers.

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b;
}

```

```

    clrscr();
    printf("enter any two values of a & b\n");
    scanf("%d %d", &a, &b);
    if (a>b)
    {
        printf("a is greater");
    }
    else
    {
        printf("b is greater");
    }
    getch();
}

```

5. WAP to compare 3 numbers and to print the largest among them without using logical Operator.

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int a,b,c;
    clrscr();
    printf("enter the a, b and c values\n");
    scanf("%d %d %d", &a, &b, &c);
    if (a>b)
    {
        if (a>c)
        {
            printf("a is greater\n");
        }
        else
        {
            printf("c is greater\n");
        }
    }
}

```

```

else if (b > c)
{
    printf("b is greater\n");
}
else
{
    printf("c is greater\n");
}
getch();
}

```

(ii) By Using Logical Operator :-

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b, c;
    clrscr();
    printf("enter the a, b & c values\n");
    Scanf("%d %d %d", &a, &b, &c);
    if (a > b & a > c)
    {
        printf("a is greater");
    }
    else if (b > c)
    {
        printf("b is greater\n");
    }
    else
    {
        printf("c is greater\n");
    }
    getch();
}

```

6. WAP to Swap two numbers by using temporary variable

```
#include <stdio.h>
#include <conio.h>
Void main()
{
    int a,b,c;
    clrscr();
    printf("enter the a & b values\n");
    Scanf("%d %d", &a, &b);
    printf("%d\n%d", a, b); // 6,5
    c = a;
    a = b;
    b = c;
    printf("%d\n%d\n", a, b); // 5,6
    getch();
}
```

7. WAP to Swap 2 numbers using temporary variable without

```
#include <stdio.h>
#include <conio.h>
Void main()
{
    int a,b;
    clrscr();
    printf("enter the a & b values\n");
    Scanf("%d %d", &a, &b);
    printf("%d\n%d\n", a, b); // 6,5
    a = a+b;
    b = a-b;
    a = a-b;
    printf("%d\n%d\n", a, b); // 5,6
    getch();
}
```

8. WAP to Swap 3 numbers take $a=10$, $b=20$, $c=5$

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

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

```
void main()
```

```
{
```

```
int a, b, c;
```

```
clrscr();
```

```
printf("enter the a, b & c values\n");
```

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

```
printf("%d %d %d", a, b, c);
```

```
a = a + b + c;
```

```
b = a - (b + c);
```

```
c = a - (b + c);
```

```
a = a - (b + c);
```

```
printf("%d %d %d", a, b, c);
```

```
getch();
```

```
}
```

9. WAP to check whether a number is odd or even

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

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

```
void main()
```

```
{
```

```
int n;
```

```
clrscr();
```

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

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

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

```
{
```

```
printf("even");
```

```
}
```

```
else,
```

```
printf("odd");
```

```
}
```

10. WAP to find whether a number is odd or even without using mod operator.

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

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

```
void main()
```

```
{
```

```
    int n;
```

```
clrscr();
```

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

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

```
if ((n/2)*2) == n)
```

```
{
```

```
    printf("even");
```

```
}
```

```
else
```

```
{
```

```
    printf("odd");
```

```
}
```

```
getch();
```

```
}
```

Note:- The return type for printf and scanf is integer

→ if (printf("Hello world\n"))

```
{
```

```
    printf("Hi"); // → Hello world
```

```
}
```

→ if (printf(" "))

```
{
```

```
    printf("Hi"); // 0.
```

```
}
```

11. WAP to find the largest among 2 numbers by using only one if condition.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b, larg;
    clrscr();
    printf("enter the a,b values\n");
    scanf("%d %d", &a, &b);
    larg = b;
    if (a > b)
    {
        larg = a;
    }
    printf("larg = %d", larg);
    getch();
}
```

12. WAP to Swap 2 numbers without using temporary variable '+' and '-'.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a, b;
    printf('enter a&b values\n');
    scanf("%d %d", &a, &b);
    printf("%d\t%d\n", a, b);
    a = a * b;
    b = a / b;
    a = a / b;
    printf("%d\t%d\n", a, b);
    getch();
}
```

13. WAP to check whether a alphabet is vowel or consonant

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

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

```
int void main()
```

```
{
```

```
    char ch;
```

```
    printf("enter single character");
```

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

```
    Switch(ch)
```

```
{
```

```
    Case 'a' :
```

```
    Case 'A' :
```

```
    Case 'e' :
```

```
    Case 'E' :
```

```
    Case 'i' :
```

```
    Case 'I' :
```

```
    Case 'o' :
```

```
    Case 'O' :
```

```
    Case 'u' :
```

```
    Case 'U' :
```

```
    printf("the Entered character is vowel\n");
```

```
    break;
```

```
    default: printf("Consonant");
```

```
}
```

```
return 0;
```

```
}
```

H. WAP to create a Simple calculator using Switch case.

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int a,b,c;
    char ch;
    clrscr();
    printf("Welcome to Simple Calculn");
    printf("enter the numbers\n");
    scanf("%d %d", &a, &b);
    printf("Select +|-*| / |%\n");
    scanf("%c", &ch);
    switch(ch)
    {
        Case '+': c = a+b;
                    printf("%d + %d = %d\n", a, b, c);
                    break;
        Case '-': c = a-b;
                    printf("%d - %d = %d\n", a, b, c);
                    break;
        Case '*': c = a*b;
                    printf("%d * %d = %d\n", a, b, c);
                    break;
        Case '/': c = a/b;
                    printf("%d / %d = %d\n", a, b, c);
                    break;
        Case '%': c = a%b;
                    printf("%d % %d = %d\n", a, b, c);
                    break;
        default: printf("Invalid option")
    }
    return 0;
}
```

Patterns :-

Nested for loop :-

(i)

```
int i, j;
for (i=0; i<4; i++)
{
    for (j=0; j<5; j++)
    {
        printf("%d\n", j)
    }
}
```

1	2	3	4	5
0	1	2	3	4
1	2	3	4	5
2	3	4	5	0
3	4	5	0	1
4	0	1	2	3

(ii) #include <stdio.h>

#include <conio.h>

void main()

```
{
    int i, j;
    for (i=0; i<4; i++)
    {
        for (j=0; j<4; j++)
        {
            printf("*\t");
        }
    }
}
```

*	*	*	*
*	*	*	*
*	*	*	*
*	*	*	*

printf("\n");

```
{
    getch();
}
```

(iii)

```
int i, j;
for (i=0; i<4; i++)
{
    for (j=1; j<=6; j++)
    {
        printf("%d\t", j);
    }
    printf("\n");
}
getch();
```

1	2	3	4	5	6
1	2	3	4	5	6
1	2	3	4	5	6
2	3	4	5	6	
2	3	4	5	6	

(iv) `for(i=1; i<=5; i++)
 {
 for(j=0; j<=4; j++)
 {
 printf("%d\t", i);
 }
 printf("\n");
 }`

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

(v) `Void main()
{
int i;
char j;
for(i=0; i<4; i++)
{
for(j='A'; j<'E'; j++)
{
printf("%c\t", j);
}
printf("\n");
}
getch();
}`

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

(vi) `for(i='A'; i<'E'; i++)
{
for(j=0; j<4; j++)
{
printf("%c\t", i);
}
printf("\n");
}`

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

```

5. for (i = A; i <= 'E'; i++)
{
    for (j = 'A'; j <= 'E'; j++)
    {
        if (i <= j)
        {
            printf("%c\t", j);
        }
        else
        {
            printf("  ");
        }
    }
}

```

A	B	C	D
A	B	C	
A	B		
A			

```

6. int i, j;
for (i = 0; i < 5; i++)
{
    for (j = 0; j < 5; j++)
    {
        if ((i + j) == 5)
        {
            printf("*\t");
        }
        else
        {
            printf("  ");
        }
    }
    printf("\n");
}

```

0	1	2	3	4	5
1	2	3	4	5	6
2	3	4	5	6	7
3	4	5	6	7	8
4	5	6	7	8	9
5	6	7	8	9	10

```

7. for (i = 0; i < 5; i++)
{
    for (j = 0; j < 5; j++)
    {
        if ((i + j) >= 5)
        {

```

```
    printf("*\t");
}
else {
    printf(" ");
}
printf("\n");
}
```

```
8. for (i=0; i<5 ; i++)
{
    for (j=0; j<5 ; j++)
    {
        if ((i+j) <= 5)
        {
            printf("*\t");
        }
        else
        {
            printf(" ");
        }
    }
    printf("\n");
}
```

*
* * *
* * * *
* * * * *
* * * * *

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

*
* * * * *
* * * * *
* * * * *
* * * * *

```

    *
  * * *
* * * * *
* * * * *
* * * * *

```

```

#include <stdio.h>
#include <conio.h>
Void main();
{
    int spc=4; str=1; i, j;
    for(j=0; j<5; j++)
    {
        for(i=0; i<spc; i++)
        {
            printf(" ");
        }
        for(i=0; i<str; i++)
        {
            printf("*");
        }
        spc--;
        str+=2;
        printf("\n");
    }
}

```

(or)

```

int i, j, n;
for(i=0; i<n; i++)
{
    for(j=0; j<n-1-i; j++)
    {
        printf(" ");
    }
    for(j=0; j<i+1+i; j++)
    {
        printf("*");
    }
    printf("\n");
}

```

	Space	star
i	n-1-i	n+i
0	4	3
1	3	2
2	2	1
3	1	0

```

3 3 3
5 5 5 5 5
7 7 7 7 7 7
9 9 9 9 9 9 9
#include <stdio.h>
#include <conio.h>
Void main();
{
    int n=5;
    int z=1;
    int i, j, k;
    clrscr();
    for(i=1; i<=n; i++)
    {
        for(j=n-1; j>=i; j--)
        {
            printf(" ");
        }
        for(k=1; k<=z; k++)
        {
            printf("%d", z);
        }
        z+=2;
        printf("\n");
    }
    getch();
}

```

3.

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

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int wid = 7
    int spc = wid/2;
    int height = wid - spc;
    int i, j, k;
    clrscr();
    for(i=height; i>=1; i--)
    {
        for(j = spc; j>=i; j--)
        {
            printf(" ");
        }
    }
    for(k=1; k<=wid; k++)
    {
        printf("%d", i);
    }
    width -= 2;
    printf("\n");
}
 getch();
}
```

4.

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

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n=5; //size
    int z=1;
    int i, j, k;
    clrscr();
    for(i=1; i<=n; i++)
    {
        for(j=n-1; j>=i; j--)
        {
            printf(" ");
        }
        for(k=0; k<z; k++)
        {
            printf("%d", i);
        }
        z+=2;
        printf("\n");
    }
    getch();
}
```

5 D
 D C
 D C B
 D C B A
 D C B
 D C
 D ~~char~~ ~~char~~
~~#include <stdio.h>~~
~~#include <conio.h>~~
 void main()
 {
 int size = 3;
 int i, j;
 clrscr();
 for (i = size; i >= -size; i--)
 {
 for (j = size; j >= abs(i); j--)
 {
 printf("%c", j + 65);
 }
 printf("\n");
 }
 getch();

6. 2 2
 3 3 3
 4 4
 5 5
~~<stdio.h>~~
~~#include <conio.h>~~
~~int~~
 void main()
 {
 int n = 5;
 int i, j;
 int px = n;
 int py = n;
 clrscr();
 for (i = 1; i < n; i++)
 {
 for (j = 1; j <= n * 2; j++)
 {
 if (j == px || j == py)
 {
 printf("%d", i);
 }
 else
 {
 printf(" ");
 }
 px--;
 py++;
 printf("\n");
 }
 return 0;

```

B   B
C   C
D   D
E   E
#include <stdio.h>
#include <conio.h>
void main()
int main()
{
    int n=5;
    int px=n;
    int py=n;
    int i,j;
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=n*2; j++)
        {
            if(j==px || j==py)
            {
                printf("%c", i+64);
            }
            else
            {
                printf(" ");
            }
        }
        px--;
        py++;
        printf("\n");
    }
    return 0;
}

```

* * * * *

```

#include <stdio.h>
#include <conio.h>
int main()
{
    int n=4;
    int px=n;
    int py=n;
    int i,j;
    clrscr();
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=n*2; j++)
        {
            if(j==px || j==py)
            {
                printf("*");
            }
            else
            {
                printf(" ");
            }
        }
        px--;
        py++;
        printf("\n");
    }
    return 0;
}

```

9.

```

    *
   * * *
  *   * *
 *     *
  *   * *
   * * *
    *

```

```

#include <stdio.h>
#include <conio.h>
int main()
{
    int n=5;
    int px=n/2+1;
    int i,j;
    clrscr();
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
        {
            if(j==px || j==n-px+1)
            {
                printf("*");
            }
            else
            {
                printf(" ");
            }
        }
        if(i<=n/2)
            px--;
        else
            px++;
        printf("\n");
    }
    return 0;
}

```

10.

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

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int m=1;
    int height=4;
    int spc=height-1;
    int i,j,k;
    clrscr();
    for(i=0;i<height;i++)
    {
        for(j=spc;j>i;j--)
        {
            printf(" ");
        }
        for(k=0;k<m;k++)
        {
            printf("%c",i+65);
        }
        m+=2;
        printf("\n");
    }
    getch();
}

```

Programs:-

1. WAP to print all natural numbers from 1 to 100. & to print from 100 to 1

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i;
    for(i=0; i<=100; i++)
    {
        printf("%d\n", i);
    }
}
```

2. WAP to print A to Z and Z to A

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char ch;
    for(ch='A'; ch<=Z; ch++)
    {
        printf("%c\n", ch);
    }
}
```

3. WAP to print all even numbers 1 to 100.

```
void main()
{
    int i;
    for(i=1; i<=50; i++)
    {
        printf("%d", i*2);
    }
}

int i;
for(i=2; i<=100; i++)
{
    if(i%2 == 0)
    {
        printf("%d\n", i);
    }
}
```

4. WAP to print all odd numbers between 1 to n.

```
void main()
{
    int i;
    for(i=1; i<=n; i+=2)
    {
        printf("%d", i);
    }
}
```

5. WAP to find the sum of all natural numbers between 1 to 10.

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

6. WAP to print (first n) sum of first n natural numbers.

```
void main()
{
    int n, sum;
    read n;
    sum = (n * (n+1)) / 2;
    printf("%d", sum);
}
```

7. WAP to find sum of all even numbers between 1 to 102

```
void main()
{
    int i, n, sum = 0;
    for (i = 2; i <= 102; i += 2)
    {
        sum = sum + i;
    }
    printf("%d", sum);
```

8. WAP to print the multiplication table 5

```
void main()
{
    int i, n;
    printf("enter n\n");
    scanf("%d", &n);
    for (i = 1; i <= 10; i++)
    {
        printf("%d * %d = %d\n", n, i, n * i);
    }
    printf("\n");
}
```

(iii) By using while :-

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

Pointers

Pointer :-

Def :- Pointer is a variable which will store the address of another variable.

```
int i=10;
int *r = &i;
int **s = &r;
int ***h = &s;
int ****a = &h;
```

i 10

FFF0

r FFF0

FFF2

s FFF2

FFF4

h FFF4

FFF6

a FFF6

FFF8

n FFF8

FFF0

h FFF4

FFF2

*s FFF0

FFF2

**s FFF0

FFF2

***s FFF0

FFF2

****s FFF0

FFF2

*****s FFF0

FFF2

printf("%d", i); // 10

printf("%p", &i); // FFF0

printf("%p", r); // FFF0

printf("%d", *r); // 10

printf("%d", **s); → invalid
indirectional error

→ h // FFF4

*h // FFF2

**h // FFF0

***h // 10

→ ****s // 10

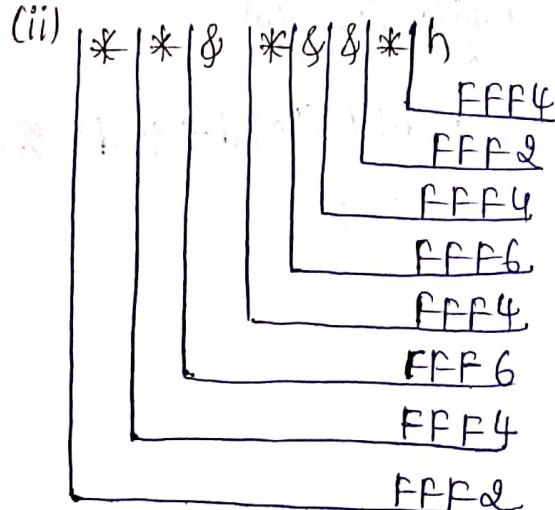
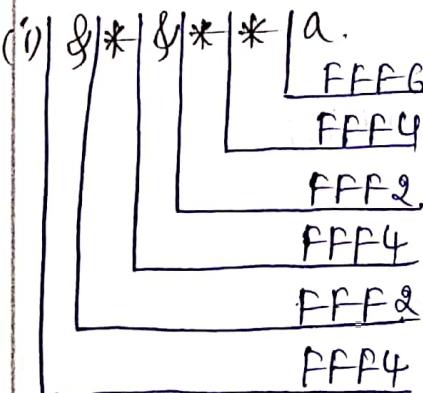
*****s // FFF0

*****s // FFF2

*****s // FFF4

*****s // FFF6

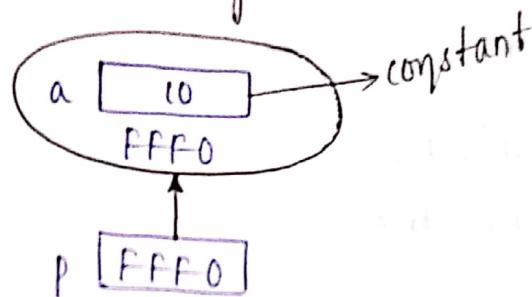
Ex:-



Constant keyword:-

Ex:- int Const a=10;

1. Pointer to a Constant integer:-



int a=10;

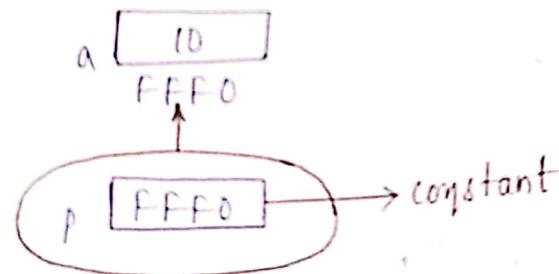
int const *p = &a;

p++; ✓

a++; ✓

(*p)++; ✗

2. Constant pointer to a Integer:-



int a=10;

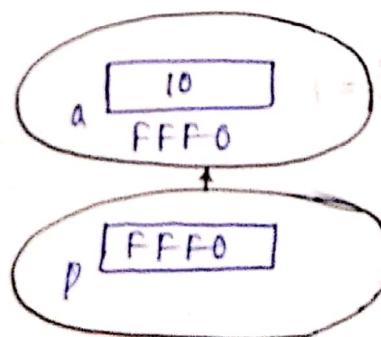
int *const p = &a;

p++; ✗

(*p)++; ✓

a++; ✓

3. Constant pointer to a Constant integer:-



int a = 10;

int const * const p = &a;

p++; ✗

(*p)++; ✗

a++; ✓

(i) #include <stdio.h>
#include <conio.h>
int a = 10;
int *p = &a;
void main()
{
 int a = 20;
 printf("%d", a); // 20
 printf("%d", *p); // 10
}

(ii) int a=1, b=0;
if(a || b++)
{
 printf("%d", a); // a=1
}
printf("%d", b); // b=0

(iii) int a=1, b=0;
if (a && b++)
{
 printf("%d", a); // a=1, value will be print
}
printf("%d", b); // b=1

MACROS (replace)

Macros :-

Syntax :-

```
| #define (macro name) (macro value) |
```

(i) `#include <stdio.h>`
`#include <conio.h>`
`#define Pi 3.14`
`void main()`
`{`
 `float r, area;`
 `r = 10;`
 `area = Pi * r * r;`
 `printf("%f", area);`
`}`

(ii) `#include <stdio.h>`
`#include <conio.h>`
`#define P 5 + 7 + 8`
`void main()`
`{`
 `int I = 5;`
 `I = I * P;`
 `printf("%d", I); // 40`
`}`

$$\begin{array}{c}
 \text{BODMAS} \\
 \downarrow \\
 I * P \\
 5 * 5 + 7 + 8 \\
 25 + 7 + 8 \\
 40
 \end{array}$$

(iii) `#include <stdio.h>`
`#include <conio.h>`
`#define begin main`
`void begin()`
`{`
 `// Statements //`
`}`

```

(iv) #include <stdio.h>
#include <conio.h>
#define f(y) y*y
void main()
{
    int a = 8;
    a = a/f(2);
    printf("%d", a); //8
}

```

$$\begin{aligned}
 9/f(2) &= 8/2 \\
 &= 4*2 \\
 &= 8
 \end{aligned}$$

v) #include <stdio.h>

```

#include <conio.h>
#define OB C
#define CB )
#define SC ;
#define OFB {
#define CFB }
#define PLUS +
#define B main
#define C ,
Void B OB CB
OFB

```

int acbcd SC

printf OB "enter the number" CB SC

scanf OB "%d %d" C &a C &b CB SC

d = a PLUS b SC

printf OB "%d" c d CB SC

CFB

ARRAYS

array :-

→ A collection of homogeneous elements stored in continuous memory.

Types of arrays

* One dimensional array

* Multi-dimensional arrays (2 to 12D)

1-D array :-

Syntax :-

```
|-----|  
| Datatype arrayname [size]; |  
|-----|
```

```
(i) #include <stdio.h>
#include <conio.h>
main()
{
    int a[5];
    a[5] = {1, 2, 3, 4, 5};
```

1	2	3	4	5
0	1	2	3	4

(iii) By using Index

```
#include <stdio.h>
#include <conio.h>
{
    int a[5];
    a[0] = 1;
    a[1] = 2;
    a[2] = 3;
    a[4] = 4;
    a[5] = 5;
```

```
(ii) #include <stdio.h>
#include <conio.h>
main()
{
    int a[5];
    a[5] = {1, 2, 3};
```

1	2	3	0	0
0	1	2	3	4

(iv) To print

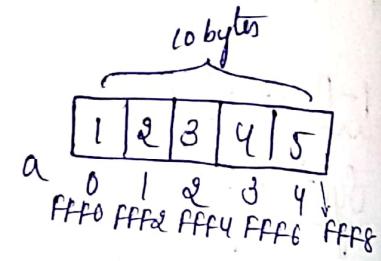
```
#include <stdio.h>
#include <conio.h>
main()
{
    int a[5];
    a[5] = {1, 2, 3, 4, 5};
    printf("%d", a[3]);
```

```

'i) #include <stdio.h>
#include <conio.h>
main()
{
    int a[5];
    printf("enter the array elements\n");
    for(i=0; i<5; i++)
    {
        scanf("%d", &a[i]);
    }
}

```

vi) #include <stdio.h>
 #include <conio.h>
 main()
 {
 int a[5];
 printf("enter the array elements\n");
 for(i=0; i<5; i++)
 {
 printf("%d", a); // FFF0
 }
 }



vii) #include <stdio.h>
 #include <conio.h>
 main()
 {
 int a[5];
 printf("enter the array elements\n");
 for(i=0; i<5; i++)
 {
 scanf("%d", a+i); // address
 }
 for(i=0; i<5; i++)
 {
 printf("%d", *a+i);
 }
 }



Application :-

Pointer to 1-D array :-

$$*(a+0) = 1;$$

$$*(a+1) = 2;$$

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

$$*(a+3) = 4;$$

$$*(a+4) = 5;$$

a	1	2	3	4	5
0	FFFF0	FFFF2	FFFF4	FFFF6	FFFF8

Multi-dimensional array :-

2-D array :- array inside another array in series.

Syntax :-

```
| datatype arrayname [size1][size2]; |
```

Ex :- int a[3][4];

0				1				2			
0	1	2	3	0	1	2	3	0	1	2	3
10	20	30	40	50	60	70	80	90	100	110	120

0 FFF0

1 FFF8

2 FFA0

$$a[0][0] = 10; \quad a[1][0] = 50;$$

$$a[2][0] = 90;$$

$$a[0][1] = 20; \quad a[1][1] = 60;$$

$$a[2][1] = 100;$$

$$a[0][2] = 30; \quad a[1][2] = 70;$$

$$a[2][2] = 110;$$

$$a[0][3] = 40; \quad a[1][3] = 80;$$

$$a[2][3] = 120;$$

application

Pointer to 2D array :-

$$\rightarrow *(*a+0)+0 = 10;$$

$$\rightarrow *(*a+1)+2 = 90;$$

$$\rightarrow *(*a+2)+3 = 65;$$

(i) `for (i=0; i<3; i++)`

{ `for (j=0; j<4; j++)`

{ `scanf("%d", (a+i)+j);`

}

}

(ii) `for (i=0; i<3; i++)`

{

`for (j=0; j<4; j++)`

{

`printf("%d", (*(*a+i)+j));`

}

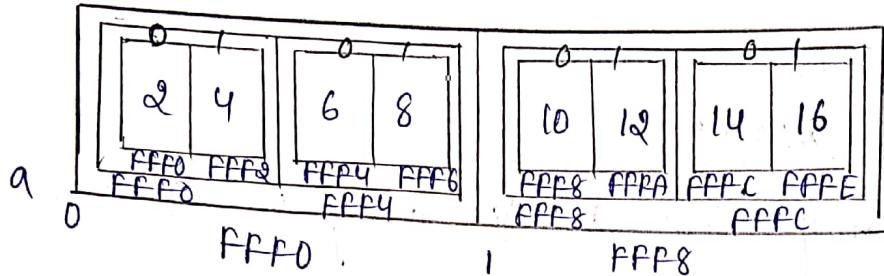
}

3D array :-

Syntax :-

datatype arrayname [size₁][size₂][size₃];

Ex :- int a[2][2][2];



$$\begin{aligned}a[0][0][0] &= 2; & a[1][0][0] &= 10; \\a[0][0][1] &= 4; & a[1][0][1] &= 12; \\a[0][1][0] &= 6; & a[1][1][0] &= 14; \\a[0][1][1] &= 8; & a[1][1][1] &= 16;\end{aligned}$$

To print 3D array elements :-

#include <stdio.h>

#include <conio.h>

main()

```
{\n    int a[2][3][2];\n    for (i=0; i<2; i++) {\n        for (j=0; j<3; j++) {\n            for (k=0; k<2; k++) {\n                printf("%d ", (*(*(*a+i)+j)+k));\n            }\n        }\n    }\n}
```

```

(ii) for(i=0; i<2; i++)
{
    for(j=0; j<3; j++)
    {
        for(k=0; k<2; k++)
        {
            scanf("%d", &((a+i)+j)+k);
        }
    }
}

```

application :- $*(*(*(*a+i)+j)+k)$

$$\rightarrow *(*(*(*a+0)+0)+1) = 4;$$

$$\rightarrow *(*(*(*a+0)+1)+0) = 6;$$

$$\rightarrow *(*(*(*a+1)+0)+1) = 12;$$

$$\rightarrow *(*(*(*a+1)+1)+0) = 14;$$

4D array :-

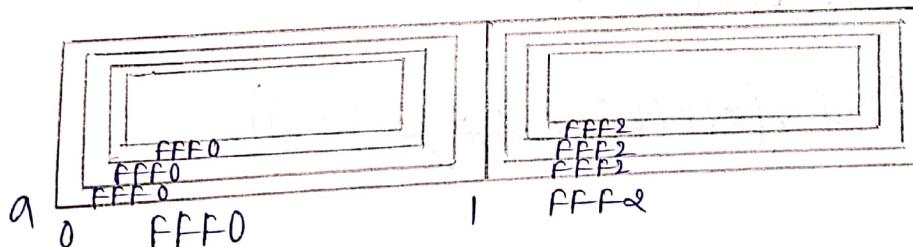
Syntax :-

```

datatype arrayname [size1][size2][size3][size4];

```

Ex :- int a[2][1][1][1];



(i) #include <stdio.h>

#include <conio.h>

void main()

```

    {
        int a[i][j][k][l];
    }

```

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

```

    {

```

```

for(j=0; j<1; j++)
{
    for(k=0; k<1; k++)
    {
        for(l=0; l<1; l++)
        {
            scanf("%d", &((a+i)+j)+k)+l);
        }
    }
}

```

(ii) #include <stdio.h>

```

#include <conio.h>
Void main()
{
    int a[i][j][k][l];
    for(i=0; i<2; i++)
    {
        for(j=0; j<1; j++)
        {
            for(k=0; k<1; k++)
            {
                for(l=0; l<1; l++)
                {
                    printf("%d", &(*(*(*(*a+i)+j)+k)+l));
                }
            }
        }
    }
}

```

- Programs :-
- WAP to read and print an integer array.
- ```
#include <stdio.h>
#include <conio.h>
void main()
{
 int arr[5];
 clrscr();
 printf("enter the array elements\n");
 for(i=0; i<5; i++)
 {
 scanf("%d", &a[i]);
 }
 printf("array elements are\n");
 for(i=0; i<5; i++)
 {
 printf("%d\n", *(&a[i]));
 }
 getch();
}
```
- WAP to print all -ve elements in an array.
- ```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a[5];
    clrscr();
    printf("enter the array elements\n");
    for(i=0; i<5; i++)
    {
        scanf("%d", &a[i]);
    }
    printf("-ve elements are\n");
```

```
for (i=0; i<5; i++)
{
    if (a[i]<0)
    {
        printf("%d\n", a[i]);
    }
    getch();
}
```

3. WAP to find the sum of all the array elements.

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

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

```
#define SIZE 100
```

```
void main()
```

```
{
    int a[SIZE], n, i, sum=0;
    clrscr();
}
```

```
printf("enter the array limit\n");
scanf("%d", &n);
```

```
printf("Enter the array elements\n");
for (i=0; i<n; i++)
{
    
```

```
    scanf("%d", &a[i]);
}
```

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

```
    sum = sum + a[i];
}
```

```
    printf("%d", sum);
}
```

```
    getch();
}
```

4. WAP to add the elements of integer array and store the result in 3rd array, finally print the 3rd array result

```
#include <stdio.h>
#include <conio.h>
#define SIZE 100
void main()
{
    int a[SIZE], i, j, n, b[SIZE], c[SIZE];
    clrscr();
    printf("enter the array limit\n");
    scanf("%d", &n);
    printf("Enter the array elements\n");
    for(i=0; i<n; i++)
    {
        scanf("%d %d", &a[i], &b[i]);
    }
    for(i=0; i<n; i++)
    {
        c[i] = a[i] + b[i];
    }
    for(i=0; i<n; i++)
    {
        printf("%d\n", c[i]);
    }
    getch();
}
```

5. WAP to find the max & min values of an integer array.

```
#include <stdio.h>
#include <conio.h>
#define SIZE 100
void main()
{
```

FUNCTIONS

functions :-

Def :- Functions are nothing but the block of codes which are written by the programmer for 'reusability' and 'modularity'.

```
1. #include <stdio.h>
#include <conio.h>
void main()
{
    clrscr();
    add(10, 20);
    add(50, 100);
    add(2, 3);
    add(1000, 10000);
    getch();
}

void add(int a, int b)
{
    int c;
    c = a + b;
    printf("%d", c); // 30
}
```

Types of functions :-

1. function without return type and without parameter
2. function without return type and with parameter
3. function with return type and without parameter
4. function with return type and with parameter
5. Recursion

R	P
X	X
X	✓
✓	X
✓	✓

function without return type and without Parameters:-

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

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

```
void add();
```

```
{
```

```
clrscr();
```

```
add();
```

```
add();
```

```
add();
```

```
getch();
```

```
}
```

```
Void add()
```

```
{
```

```
int a,b,c;
```

```
printf("enter the values of a&b");
```

```
Scanf("%d%d", &a, &b);
```

```
Printf("%d", c);
```

```
}
```

2. function without return type and with parameter

```
#include <stdio.h>
#include <conio.h>
Void add(int, int)
Void main()
{
    clrscr();
    add(10, 20);
    getch();
}

Void add(int a, int b)
{
    int c;
    c = a + b;
    printf("%d", c);
}
```

3. function with return type and without parameter

```
#include <stdio.h>
#include <conio.h>
int add();
Void main()
{
    clrscr();
    printf("%d", add());
    printf("%d", add());
    getch();
}

int add()
{
    int a,b,c;
    printf("enter the values of a & b");
    Scanf("%d%d", &a, &b);
    c = a+b;
    return c;
}
```

4. function with return type and with parameter

```
#include <stdio.h>
#include <conio.h>
int add(int, int);
void main()
{
    clrscr();
    printf("%d", add(10, 20));
    printf("%d", add(50, 80));
    getch();
}

int add(int a, int b)
{
    return a+b;
}
```

How to pass an array as argument

```
#include <stdio.h>
#include <conio.h>
void print (int a[], int);
void main()
{
    int a[5] = {1, 2, 3, 4, 5};
    int size = Size of (a) / Size of (*a);
    clrscr();
    print (a, size);
}
void print (int c[], int n)
{
    int i;
    for (i=0; i<n; i++)
    {
        printf ("%d", c[i]);
    }
}
```

Methods of Calling :-

1. Call by Value :-

```
#include <stdio.h>
#include <conio.h>
void mul (int c, int d);
{
    int e;
    e = a * b;
    printf ("%d", e);
}
void main()
```

```
int a=10, b=5;  
clrscr();  
mul(a, b);  
getch();
```

(ii) Call by address :-

```
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
    int a=10, b=5;  
    clrscr();  
    mul(&a, &b);  
    getch();  
  
    void mul(int *c, int *d)  
    {  
        int e;  
        e = (*c) * (*d);  
        printf("%d", e);  
    }  
}
```

Programs :-

1. WAP to find a number is odd or even by using function

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

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

```
void oddeven(int n);
```

```
void main()
```

```
{  
    int a=10, b=5;
```

```
    clrscr();
```

```
    odd even(5);
```

```
    odd even(6);
```

```
    getch();
```

void oddeven (int a)

```
{  
    if (a%2 == 0)  
    {  
        printf("even");  
    }  
    else  
    {  
        printf("odd");  
    }  
}
```

(ii) To print 012343210 :-

```
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
    int i;  
    for(i=0; i<=4; i++)  
    {  
        printf("%d", i);  
    }  
    for(i=3; i>=0; i--)  
    {  
        printf("%d", i);  
    }  
}
```

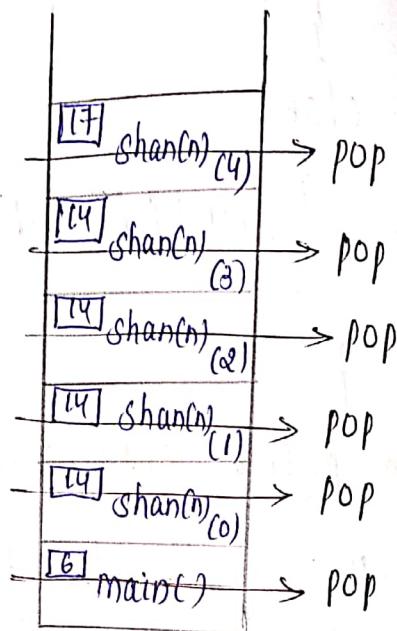
* ; → It terminates and it save the memory.

* if ; is not used it prints continuously in the memory.

5. Recursion :-

Def:- A function calling by itself is called Recursion.

```
(i) #include <stdio.h>
#include <conio.h>
Void shan (int); → 1
int n; → 2
Void main() → 3
push ← {
    clrscr(); → 4
    Shan (n); → 5
    getch(); → 6
    → 7
}
Void Shan(int n) → 8
{
    → 9
    if (n < 4) → 10
    {
        → 11
        printf("%d", n); → 12
        Shan (n+1); → 13
        → 14
        printf("%d", n); → 15
    } → 16
} → pop → 17
```



0 1 2 3 4 5 6 7 8 9 10

(i) WAP to find factorial of a number

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

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

```
Void main()
```

```
{
    int n, a, fact = 1;
    printf("enter the n value\n");
    Scanf("%d", &n);
    for (a = 1; a <= n; a++)
    {
        fact = fact * a;
    }
```

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

```
}
```

ii) WAP for factorial of a number by using function.

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

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

```
void fact(int)
```

```
int n;
```

```
void main()
```

```
{
```

```
clrscr();
```

```
fact(6);
```

```
} getch();
```

```
void fact(int n)
```

```
{
```

```
int i, fact = 1;
```

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

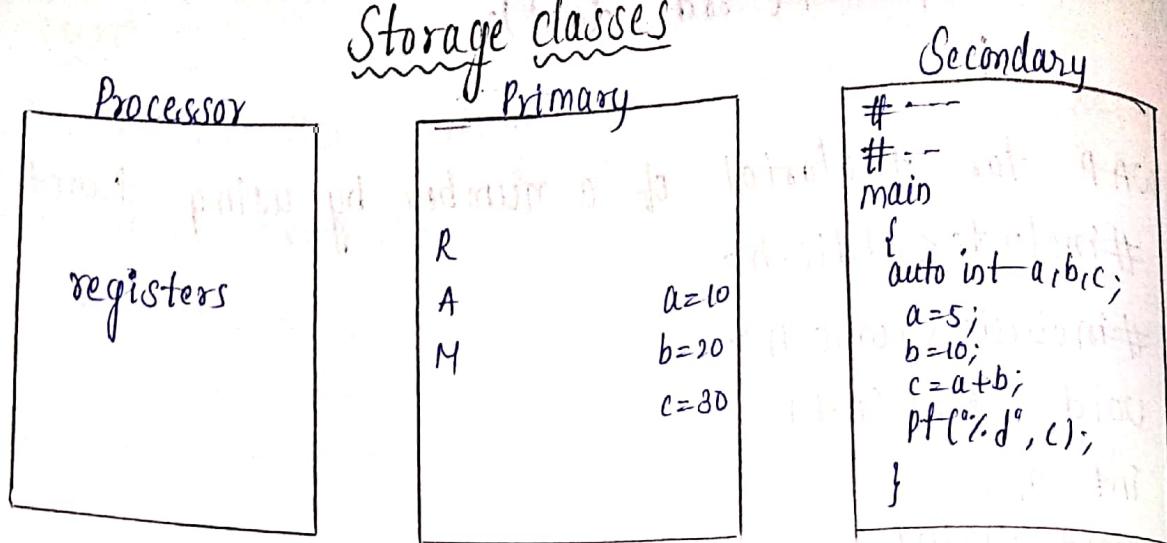
```
{
```

```
    fact = fact * i;
```

```
}
```

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

```
}
```



Storage classes are 4 types.

1. auto
2. register
3. static
4. extern

1. auto :- (Local Variable type)

default value : Garbage value

Life : Till the end of the block

Scope : Within the block.

Storage : RAM.

2. register :-

default value : Garbage value

Life : Till the end of the block

Scope : Within the block

Storage : CPU

3. static :-

default value : 0

Life : Till the end of the program

Scope : Within the block.

Storage : Data Segment

Dynamic Memory allocation :-

→ Allocating memory dynamically is known as dynamic memory allocation.

(i) malloc()

(ii) calloc()

(iii) realloc()

(iv) free()

(i) malloc() :-

#include <stdlib.h>

→ Malloc is used to allocate memory dynamically for structures.

→ Note :- The default return type for malloc() & calloc are Type casting :- changing 1 datatype to another datatype.

(i) int *ptr;

ptr = (int *) malloc(10 * size of (int)); ^{we can change the value.}

(ii) calloc() :-

→ Here we can create multiple blocks

→ It is used for arrays

(i) int *ptr;

ptr = (int *) calloc(10, 4 * size of (int));

(iii) realloc() :-

*ptr = realloc(ptr, 3 * size of (int));

*ptr = realloc(ptr, 10 * size of (int));

(iv) Free() :-

*Free(ptr);

CommandLine arguments :-

(i) `#include <stdio.h>`
`#include <conio.h>`
`#include <stdlib.h>`

```
int main (int argc, char *argv[])
{
    int i;
    clrscr();
    for(i=0; i<argc; i++)
    {
        printf("In argv[%d] = %s", i, argv[i]);
    }
    return 0;
}
```

→ When IDE's are not there we take another alternatives
is known as command line arguments

→ we can write program by using Notepad.

→ IDE → Integrated developer environment.

(ii) To add 2 numbers by using Command line arguments.

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

int main (int argc, char *argv[])
{
    int a, b, c, i;
    clrscr();
    a = atoi(argv[1]);
    b = atoi(argv[2]);
    c = a+b;
    printf("%d", c);
    return 0;
}
```

Programs :-

Q. WAP to find the sum of digits in a given number

```
#include<stdio.h>
#include <conio.h>
void main()
{
    int n, r, sum=0;
    printf("Enter a value\n");
    scanf("%d", &n);
    do
    {
        r = n % 10;
        sum = sum + r;
        n = n / 10;
    } while (n > 0);
    printf("The sum of digits of %d is %d", n, sum);
}
```

Q. WAP to check a number is pallindrome or not.

```
#include<stdio.h>
#include <conio.h>
int main()
{
    int n, t, reverse = 0;
    printf("Enter a number to check if it is pallindrome\n");
    scanf("%d", &n);
    t = n;
    while (t != 0)
    {
        reverse = reverse * 10;
        reverse = reverse + t % 10;
    }
}
```

```

} t = t/10;
if (n == reverse)
    printf("%d is a palindrome number(%d, n));
else
    printf("%d isn't a palindrome number(%d, n);
return 0;
}

```

3. WAP to check whether a number is even or odd by using switch case.

```

#include <stdio.h>
#include <conio.h>
int main()
{
    int n;
    printf("enter a the value ln");
    scanf("%d", &n);
    Switch (n%2)
    {
        Case 0 : printf("%d is an Even numberln", n);
        break;
        Case 1 : printf("%d is an odd numberln", n);
        break;
    }
    return 0;
}

```

4. WAP to find maximum between 2 numbers using Switch.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int n1, n2;
    printf("enter 2 numbers to find maximum");
    scanf("%d %d", &n1, &n2);
    switch (n1 > n2)
    {
        case 0: printf("%d is maximum", n2);
        break;
        case 1: printf("%d is maximum", n1);
        break;
    }
    return 0;
}
```

5. WAP to check whether a number is prime or not by using Switch.

6. To read and print the given number in reverse order

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int n, reverse = 0;
    printf("Enter a number to reverse\n");
    scanf("%d", &n);
    while (n != 0)
    {
        reverse = reverse * 10;
        reverse = reverse + n % 10;
        n = n / 10;
    }
}
```

```
    } // End of function reverse

    printf("reverse of entered number is = %d\n", reverse);
    return 0;
} // End of main function
```

```
5. #include <stdio.h>
#include <conio.h>
void main()
{
    int n;
    printf("enter the number\n");
    scanf("%d", &n);
    for (i=2; i<n; i++)
    {
        if (n%i==0)
        {
            flag=0;
            break;
        }
        flag=1;
    }
    switch(flag)
    {
        case 0 : printf(" prime number");
                    break;
        case 1 : printf(" not prime number");
                    break;
    }
}
```

Structures and Unions

Structure :-

Def:- Structures are nothing but the collection of heterogeneous elements stored in continuous memory.

Syntax :-

```
Struct structurename  
{  
    datatype, variable;  
    datatype, variable;  
    ;  
    ;  
    ;  
    datatype n variable;  
};
```

```
(i) #include <stdio.h>  
#include <conio.h>  
Struct std  
{  
    int id;  
    Char name[50];  
    float totalm;  
};  
Void main()  
{  
    char a[50] = "Ishan";  
    Struct std x;  
    x.id = 10;  
    x.totalm = 512;  
    strcpy(x.name, a);  
    printf("%d", x.id);  
    printf("%d", x.totalm);  
    printf("%s", x.name);  
}
```

(ii) To store the details of an Employee

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

struct employ
{
    int id;
    char name[20];
    float salary;
};

Void main()
{
    char a[20];
    clrscr();
    printf("Enter the id\n");
    scanf("%d", &x.id);
    printf("Enter the name\n");
    printf("Enter the name\n");
    Scanf("%s", a);
    strcpy(x.name, a);
    printf("Enter the Salary\n");
    Scanf("%d", &x.sal);
    printf("%d", x.id);

    printf("%s", x.name);
    printf("%.2f", x.sal);
    getch();
}
```

(iii)

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

struct emp
{
    int id;
    char name[20];
```

```

float Sal;
};

Void main()
{
    struct emp x[100];
    char a[20];
    int i;
    clrscr();
    for(i=0; i<100; i++)
    {
        printf("Enter the id\n");
        scanf("%d", &x.id);
        for(i=0; i<100; i++)
        {
            printf("Enter the name\n");
            scanf("%s", a);
            for(i=0; i<100; i++)
            {
                printf("Enter the salary\n");
                scanf("%f", &x.Sal);
            }
        }
    }
}

```

Pointer to a Structure :-

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
struct emp
{
    int id;
    char name[20];
    float Sal;
} x;

```

x	2	20	4
---	---	----	---

```

int a;
int *p = &a;
a = 10;
*p = 10;

```

```
void main()
{
    struct emp *p = &x; // no place will go to
    x.id = 10;
    p->id = 10;
    p->Sal = 1000;
}
```

'->' = indirection operator

Structure inside another structure :-

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
Struct std
{
    int id;
    char name[20];
}
Struct mark
{
    float m1, m2, m3, m4;
    } y;
    } x;
```

```
main()
{
    x.id = 10;
    x.y.m1 = 36;
    x.y.m2 = 70;
    x.y.m3 = 90;
    x.y.m4 = 99;
}
```

Onions :-

→ at a time only one value can be stored and printed

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

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

```
#include <string.h>
```

Union std

{

~~int~~ id;

```
char name[20];
```

float totalm;

}

20 bytes

Strings

String :-

Def :- An array with character datatype is called string

→ The string always end with null(\\0);

Syntax :-

```
{char Stringname [size];}
```

```
char a[7] = {'S', 't', 'r', 'i', 'n', 'g', '\\0'}; ]
```

```
a[6] = "STRING";
```

```
a[0] = 'S';
```

```
a[1] = 'T';
```

```
a[2] = 'R';
```

```
a[3] = 'I';
```

```
a[4] = 'N';
```

```
a[5] = 'G';
```

S t r i n g \\0

S T R I N G \\0

```
(i) #include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    char a[20];
    clrscr();
    printf("Enter a string\n");
    scanf("%s", a);
    printf("%s", a); // Hello
}
```

Hello world.

H e l l o w o r l d

```
(ii) void main()
{
    char a[20];
    clrscr();
    printf("Enter a string\n");
    gets(a);
    printf("%s", a); //Hello-world
}
```

H	e	l	l	o		w	o	r	l	d
---	---	---	---	---	--	---	---	---	---	---

Applications of pointer to character(string):-

(i) void main()

```
{
    char a;
    char *p = &a;
    *p = "India is my Country";
}
```

(ii) void main()

```
{
    char *p = "Signature";
```

Snippets:-

(i) char *p = "Hello";

printf("%s", p); //Hello

(ii) *p++;

printf("%s", p); //Hello

H	e	l	l	o		k		m	i	n
---	---	---	---	---	--	---	--	---	---	---

P FFFF0

(iii) printf("%c", *(p+i)); //e

(iv) printf("%s", (p+2)); //lo

*p+=2; ++;

*p+=2; --;

printf("%s", p);

1. WAP to find the length of a string by using for loop.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
main()
{
    char text[200];
    int i, len = 0;
    printf("enter the text\n");
    gets(text);
    for (i = 0; text[i] != '\0'; i++)
    {
        len = len + 1;
    }
    printf("the length is %d", len);
}
```

2. Write a program to reverse a string and to check whether a string is palindrome or not.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
int main()
{
    char str[100], tmp, rev[100];
    int begin, end;
    printf("Enter string");
    scanf("%s", &str);
    strcpy(rev, str);
    begin = 0;
    end = strlen(str) - 1;
    while (begin < end)
    {
        tmp = str[begin];
        str[begin] = str[end];
        str[end] = tmp;
        begin++;
        end--;
    }
    if (strcmp(str, rev) == 0)
        printf("String is Palindrome");
    else
        printf("String is not Palindrome");
}
```

```

        strEndJ = tmp;
        beginJ++;
        endJ--;
    }
    printf("Reverse string : %s\n", str);
    if (strcmp(str, revStr) == 0)
        printf("%s is palindrome!\n", str);
    else
        printf("%s is not palindrome!\n", str);
}
return 0;

```

WAP to convert binary to decimal.

```

#include <stdio.h>
#include <conio.h>
#include <math.h>
int ConvertBinaryToDecimal(long long n);
int main()
{
    long long n;
    printf("Enter a binary number\n");
    scanf("%lld", &n);
    printf("%lld in binary = %d in decimal", n, convertBinary(n));
    return 0;
}

int ConvertBinaryToDecimal(long long n)
{
    int decimalNumber = 0, i = 0, remainder;
    while (n != 0)
    {
        remainder = n % 10;
        n /= 10;
        decimalNumber += remainder * pow(2, i);
        i++;
    }
    return decimalNumber;
}

```

4. WAP for GCD - recursion

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

int gcd(int a, int b)
{
    if (a % b == 0)
        return b;
    else
        return gcd(b, a % b);
}

void cat(char *p, char *q)
{
    while (*p)
        p++;
    while (*p++ = *q++);
}

int main(int argc, char *argv[])
{
    int i;
    for (i = 0; i < argc; i++)
    {
        printf("In argv[%d] = %s", i, argv[i]);
    }
    getch();
    return 0;
}

```

5. WAP for fibonacci Series

```

#include <stdio.h>
#include <conio.h>

int main()
{
    int i, range;
    long int arr[40];
    printf("enter the number, range\n");

```

```

    Scanf ("%d", &range);
    arr[0] = 0;
    arr[1] = 1;
    for(i=2; i<range; i++)
    {
        arr[i] = arr[i-1] + arr[i-2];
    }
    printf("fibonacci series is\n");
    {
        for(i=0; i<range; i++)
        {
            printf("%d", arr[i]);
        }
        return 0;
    }
}

```

Armstrong

```

#include <stdio.h>
#include <conio.h>
#include <math.h>
int main()
{
    int number, original_number, remainder, result = 0, n = 0;
    printf("Enter an integer\n");
    Scanf ("%d", &number);
    Original number = number;
    while (original number != 0)
    {
        original number /= 10;
        ++n;
    }
    original number = number;
    while (original number != 0)
    {
        remainder = original number % 10;
    }
}

```

```

        result = pow(remainder, n);
    }

    if (result == number)
        printf("%d is an Armstrong number", number);
    else
        printf("%d is not an Armstrong number", number);
    return 0;
}

```

7. Arithmetic - Progression :-

```

#include <stdio.h>
#include <conio.h>
#include <math.h>
int main()
{
    int a, d, n, i, tn;
    int sum = 0;
    printf("Enter the 1st no. of the A.P Series");
    scanf("%d", &a);
    printf("Enter the total no. of the A.P Series");
    scanf("%d", &n);
    printf("Enter the common difference of A.P Series");
    scanf("%d", &d);
    sum = (n * (2 * a + (n - 1) * d)) / 2;
    tn = a + (n - 1) * d;
    printf("Sum of the Series A.P");
    for (i = a; i <= tn; i = i + d)
    {
        if (i == tn)
            printf("%d + ", i);
        else
            printf("%d = %d", i, sum);
    }
    return 0;
}

```

8. Geometric - Progression :-

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

int main()
{
    float a, r, i, tn;
    int n;
    float sum = 0;
    printf("Enter the 1st no. of the G.P Series");
    scanf("%f", &a);
    printf("Enter the total nos of in the G.P Series");
    scanf("%d", &n);
    printf("Enter the common ratio of G.P Series");
    scanf("%f", &r);
    sum = (a * (1 - pow(r, n + 1))) / (1 - r);
    tn = a * (1 - pow(r, n - 1));
    printf("tn term of G.P %f", tn);
    printf("In sum of the G.P %f", sum);
    return 0;
}
```

9. WAP to insert an element in an array.

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

int main()
{
    printf("In In It Studgtonight");
    int array[100], position, c, n, value;
    printf("In In Enter no. of elements in array");
    scanf("%d", &n);
    printf("In In Enter %d Elements", n);
    for (c = 0; c < n; c++)
        array[c] = c;
    position = 5;
    value = 100;
    array[position] = value;
    for (c = 0; c < n; c++)
        printf("%d ", array[c]);
}
```

```

scanf("%d", &array[c]);
printf("In In Enter the location where u want to insert
new element");
scanf("%d", &position);
printf("In In Enter the value to insert");
scanf("%d", &value);
}

for (c=n-1; c>=position-1; c--)
{
    array[c+1] = array[c];
    array[position-1] = value;
}
printf("In In Resultant array is");
for (c=0; c<n; c++)
{
    printf("%d", array[c]);
}
return 0;
}

```

10. WAP for Harmonic Progression

```

#include <stdio.h>
#include <conio.h>
void main()
{
    int n;
    float i, sum, t;
    printf("1+1/2+1/3+...+1/n\n");
    printf("Enter the value of n\n");
    scanf("%d", &n);
    sum = 0;
    for (i=1; i<n; i++)
    {
        t = 1/i;
        sum = sum + t;
    }
    printf("Sum = %f", sum);
}

```

```
    printf("%f", sum);
    getch();
}
```

11. WAP to convert number from decimal to binary

```
#include <stdio.h>
int main()
{
    int number, cnt, i;
    int bin[32];
    printf("Enter decimal number\n");
    scanf("%d", &number);
    cnt = 0;
    while (number > 0)
    {
        bin[cnt] = number % 2;
        number = number / 2;
        cnt++;
    }
    printf("Binary value is\n");
    for (i = (cnt - 1); i >= 0; i--)
        printf("%d", bin[i]);
    return 0;
}
```

12. WAP to print all leap years from 1 to n.

```
#include <stdio.h>
#include <conio.h>
int checkleapyear(int year)
{
    if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0))
        return 1;
    else
        return 0;
}
```

```

int main()
{
    int i, n;
    printf("Enter the value of n\n");
    scanf("%d", &n);
    printf("leap years from 1 to %d\n", n);
    for(i=1; i<=n; i++)
    {
        if(checkleapyear(i))
            printf("%d\n", i);
    }
    return 0;
}

```

13. WAP for a complete working C program to demonstrate all insertion methods. (inserting_node_after)

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

struct Node
{
    int data;
    struct node *next;
};

void insertAfter(struct Node *prev-node, int new-data)
{
    if(prev-node == NULL)
    {
        printf("the given previous node cannot be NULL");
        return;
    }
    struct node *new-node = (struct Node *) malloc(sizeof(struct Node));
    new-node->data = new-data;
}

```

```

new_node->next = prev_node->next;
prev_node->next = new_node;
}

void printList( struct Node *node)
{
    while( node != NULL)
    {
        printf("%d", node->data);
        node = node->next;
    }
}

int main()
{
    struct Node *head = NULL;
    insertAfter(head->next, 8);
    printf("In created linked list is \n");
    printList(head);
    return 0;
}

```

3. WAP to find the min and max sum of matrix

```

#include <stdio.h>
#include <conio.h>
Void minmax( int a[], int n)
{
    int i, min, max, sum;
    min = max = sum = a[0];
    for( i=1; i<n; i++)
    {
        if( min < a[i])
            min = a[i];
        if( max > a[i])
            max = a[i];
        sum = sum + a[i];
    }
}

```

```
printf("Min Sum = %d, Max Sum = %d", sum-min, sum+max);
}
int
```

for finding sum of all substrings

14. WAP to generate substrings in a given string.

```
#include <stdio.h>
#include <conio.h>
void Substring (char str[], int n)
{
    int i, j, l, k;
    for (l=1; l<=n; l++)
    {
        for (i=0; i<=n-l; i++)
        {
            j = i+l-1;
            for (k=i; k<=j; k++)
                printf("%c", str[k]);
            printf("\n");
        }
    }
}
int main()
{
    char str[] = "abc";
    Substring (str, 3);
    getch();
    return 0;
}
```

15. WAP for finding Max & min of an array

```
#include <stdio.h>
#include <conio.h>
Struct pair
{
    int min;
    int max;
};

Struct pair getMinMax(int arr[], int n)
{
    Struct pair minmax;
    int i;
    if (n == 1)
    {
        minmax.max = arr[0];
        minmax.min = arr[0];
        return minmax;
    }
    if (arr[0] > arr[1])
    {
        minmax.max = arr[0];
        minmax.min = arr[1];
    }
    else
    {
        minmax.max = arr[1];
        minmax.min = arr[0];
    }
    for (i = 2; i < n; i++)
    {
        if (arr[i] > minmax.max)
            minmax.max = arr[i];
        else if (arr[i] < minmax.min)
            minmax.min = arr[i];
    }
    return minmax;
}
```

```

15 int main()
{
    int arr[] = {1000, 11, 445, 11330, 3000};
    int arr-size = 6;
    struct pair minmax = getMinMax(arr, arr-size);
    printf("n Minimum element is %d", minmax.min);
    printf("n Maximum element is %d", minmax.max);
    getch();
}

```

16. WAP to check Prime number.

```

#include <stdio.h>
#include <conio.h>
int main()
{
    int n, flag=0, i;
    clrscr();
    printf("Enter the number");
    scanf("%d", &n);
    for(i=2; i<=n/2; i++)
    {
        if (n%i==0)
        {
            flag=1;
            break;
        }
    }
    if (flag==0)
        printf("The number is prime");
    else
        printf("The number is not prime");
    getch();
    return 0;
}

```

17. WAP to check nth bit in a 32 bit integer is set or not

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int n,k;
    printf("enter a 32 bit number\n");
    scanf("%d", &n);
    printf("enter the bit no to check...\n");
    printf("bit-no 0-indexed & 0 starts from LSB...\n");
    scanf("%d", &k);
    if(n > 32)
    {
        printf("enter between 0-31\n");
        return;
    }
    k = k >> n;
    if(k & 1 == 1)
        printf("%d the bit is set\n", n);
    else
        printf("%d the bit not set\n", n);
    return 0;
}
```

18. WAP to find Substring in a given string.

```
#include <stdio.h>
#include <conio.h>
void Substring(char str[], int n)
{
    int i, j, k;
    for(l=1; l<=n; l++)
    {
        for(i=0; i<=n-l; i++)
        {
            . . .
        }
    }
}
```

```

        for (k=i; k<=j; k++)
            printf("%c", str[k]);
        printf("\n");
    }
}

printf("Input a string with length <= 100\n");
int main()
{
    char str[100] = "abc";
    Sub_string(str, 3);
    getch();
    return 0;
}

```

19. WAP for Panagram - of - strings. (a program to check if a string contains all the letters of English alphabet)

```

#include <stdio.h>
#include <conio.h>
void main()
{
    char s[100];
    int i, used[26]={0}, total=0;
    clrscr();
    printf("Enter the string\n");
    gets(s);
    for(i=0; s[i]!='\0'; i++)
    {
        if ('a'<=s[i]&&s[i]<='z')
            total += !used[s[i]-'a'];
        used[s[i]-'a']=1;
    }
    else if ('A'<=s[i]&&s[i]<='Z')
    {

```

```

total + = ! used [s[i]] - 'A' );
used [s[i]] = 'A';
}

if (total == 26)
{
    printf("The entered string is a pangram string");
}
else
{
    printf("The entered string is not a pangram string");
}
getch();
}

3. WAP to reverse a string.
#include <stdio.h>
#include <conio.h>
void reverse (char * begin, char * end);
void reverseWords (char * s)
{
    char * word_begin = s;
    char * temp = s;
    while (*temp)
    {
        temp++;
        if (*temp == ' ')
            reverse (word_begin, temp - 1);
        else if (*temp == ',')
            reverse (word_begin, temp - 1);
        word_begin = temp + 1;
    }
}

4. WAP to print primes in given range M and N
#include <stdio.h>
#include <conio.h>
int main()
{
    int i, j, m, n, flag;
    clrscr();
    printf("Enter the Range M and N");
    scanf("%d %d", &m, &n);
    if (m == 2)

```

```
printf("%d\n", N);
```

```
N++;
```

```
} // loop ends
```

```
if (N < 2)
```

```
{ printf("No primes");
```

```
exit(0);
```

```
} if (M%2 == 0)
```

```
{ M++;
```

```
for(i = M; i <= N; i += 2)
{ M++;
  if (i == M)
    printf(" %d", i);
  else
    printf(", %d", i);
}
```

```
flag = 0;
```

```
for(j = 3; j <= i/2; j = j + 2)
{ if (i % j == 0)
  flag = 1;
  break;
}
```

```
if (flag == 1)
  printf(" %d", i);
else
  printf(" %d", i);
}
```

```
if (*z == '\n')
  z += a;
else
  putchar(*z);
z++;
}
```

```
if (*z == ' ')
  break;
}
```

```
if (cflag == 0)
  printf("%d\n", i);
}
```

```
getch();
return 0;
}
```

Q3. WAP for reversing words in a given string
#include <csound.h>
#include <conio.h>
int getch()
{ char c;
 while ((c = _getch()) != 'q')
 if (c != ' ')
 _putch(c);
 else
 break;
 return c;
}

Q3. WAP to reverse a sentence entered by user without using strings.
#include <csound.h>
#include <conio.h>
void reverseSentence();
int main()
{
 printf("Enter a sentence");
 reverseSentence();
 return 0;
}

Q3. WAP for reversing words in a given string
#include <csound.h>
#include <conio.h>
int getch()
{ char c;
 while ((c = _getch()) != 'q')
 if (c != ' ')
 _putch(c);
 else
 break;
 return c;
}

Q3. WAP to reverse a sentence entered by user without using strings.
#include <csound.h>
#include <conio.h>
void reverseSentence();
int main()
{
 printf("Enter a sentence");
 reverseSentence();
 return 0;
}

```

24. Void reverseSentence()
{
    char c;
    Scanf("%c", &c);
    If (c != '\n')
    {
        reverse sentence();
        printf("%c", c);
    }
}

25. WAP to find out the sum of series 1 + 2 + 3 + ... + n.
#include <stdio.h>
#include <conio.h>
int main()
{
    int n, i;
    int sum = 0;
    printf("Enter the n i.e max values of series");
    Scanf("%d", &n);
    printf("Sum of");
    Sum = (n * (n + 1) * (2 * n + 1)) / 6;
    printf("Sum of the series\n");
    for (i = 1; i <= n; i++)
    {
        if (i == n)
            printf("%d + ", i);
        else
            printf("%d + %d", i, sum);
    }
    return 0;
}

```

25. String Palindrome Program

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
void isPalindrome (char str[])
{
    int l = 0;
    int h = strlen(str) - 1;
    while (h > l)
    {
        if (str[l + j] != str[h - j])
        {
            printf("%s is Not Palindrome", str);
            return;
        }
    }
    printf("%s is palindrome", str);
}

int main()
{
    isPalindrome("abba");
    isPalindrome("geeks");
    return 0;
}

26. String Concatenation
#include <stdio.h>
#include <string.h>
void concat (char[], char[]);
int main()
{
    char s1[50], s2[30];
    printf("Enter string 1");
    gets(s1);

```

```

printf("Enter string : ");
get(s1);
concat(s1, s2);
printf("Concatenated string is %s", s1);
return 0;
}

```

```

void concat(char s1[], char s2[])
{
    int i, j;
    i = strlen(s1);
    for(j=0; s2[j] != '\0'; i++, j++)
    {
        s1[i] = s2[j];
    }
    s1[i] = '\0';
}

```

Q. WAP for returning reversed string using recursion

```

#include <stdio.h>
#include <conio.h>
char *rev(char *p)
{
    static q[100];
    static int i;
    if (*p)
    {
        rev(p+1);
        q[i++] = *p;
    }
    return q;
}
int main()
{
    // Your code here
}

```

Q. WAP to find out the sum of series $1/3 + 2/3 + \dots + n/3$.

```

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

```

```

    int n, i;
    int sum = 0;

```

```

    printf("Enter the n.i.e max values of series\n");
    scanf("%d", &n);
    sum = (1 * (n+1) * (2 * n + 1)) / 6;
    printf("Sum of the series\n");
    for(i=1; i<=n; i++)
    {
        if (i == n)
            printf("%d = %d", i, sum);
        else
            printf("%d + ", i);
    }
}

```

Q. WAP to generate check sparse matrix

```
#include <stdio.h>
#include <conio.h>
int main()
{
    printf("In In It is\n");
    int n, m, c, d, matrix[10][10];
    int counter = 0;
    printf("In enter the number of rows and columns of the matrix\n");
    scanf("%d %d", &m, &n);
    printf("In Enter the %d Elements of the matrix\n", m * n);
    for (c = 0; c < m; c++)
    {
        for (d = 0; d < n; d++)
        {
            scanf("%d", &matrix[c][d]);
            if (matrix[c][d] == 0)
                Counter++;
        }
    }
    printf("In In the entered matrix is\n");
    for (c = 0; c < m; c++)
    {
        for (d = 0; d < n; d++)
        {
            printf("%d\t", matrix[c][d]);
        }
        printf("\n");
    }
    if (Counter > m * n / 2)
        printf("In In the entered matrix is a sparse matrix\n");
}
```

else
printf("In In the entered matrix is not a sparse matrix\n");
return 0;
}

Q. WAP to check two strings are anagrams

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
int main(void)
{
    char s1[] = "recitals";
    char s2[] = "articles";
    char temp;
    int i, j;
    int n = strlen(s1);
    int n1 = strlen(s2);
    if (n1 != n)
    {
        printf("%s and %s are not anagrams!\n", s1, s2);
        return 0;
    }
    for (i = 0; i < n - 1; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (s1[i] > s1[j])
            {
                temp = s1[i];
                s1[i] = s1[j];
                s1[j] = temp;
            }
            if (s2[i] > s2[j])
            {
                temp = s2[i];
                s2[i] = s2[j];
                s2[j] = temp;
            }
        }
    }
}
```

$S_2[i:j] = S_2[j:j]$;

$S_2[j:j] = temp$;

}

}

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

{

if (S1[i] != S2[i])

{

printf("strings are not anagrams! \n", S1, S2);
return 0;

}

30.

Q. WAP. to find the biggest number in an array of numbers using Recursion

#include <csdio.h>

#include <stdlib.h>

#include <conio.h>

#include <limits.h>

int max (int a, int b)

{

return (a>b)? a : b;

}

int find big Rec (int * a, int n)

{

if (n==0)

return INT_MIN;

else
return max(a[n-1], find big Rec (a, n-1));

}

int main ()

{

printf("Enter array length\n");
scanf ("%d", &n);

int * a = (int *) (malloc (sizeof (int)*n));
printf ("Enter elements -\n");

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

{

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

}

int big = find big Rec (a, n);

printf ("The biggest element in the array is %d\n",

big);

return 0;

}

}