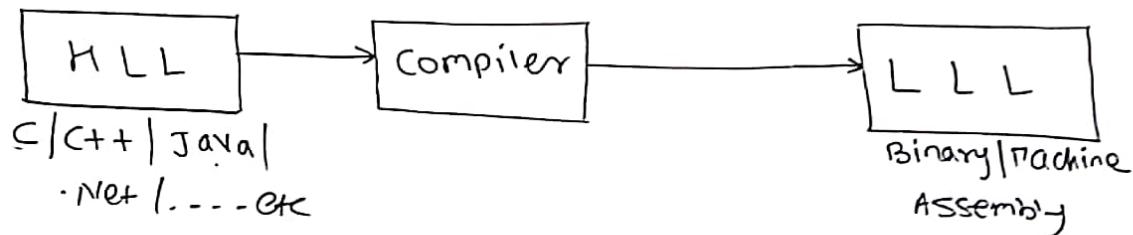
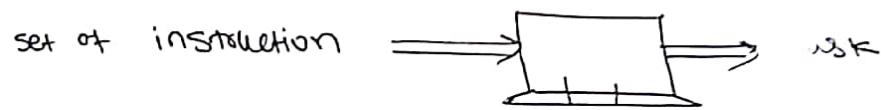
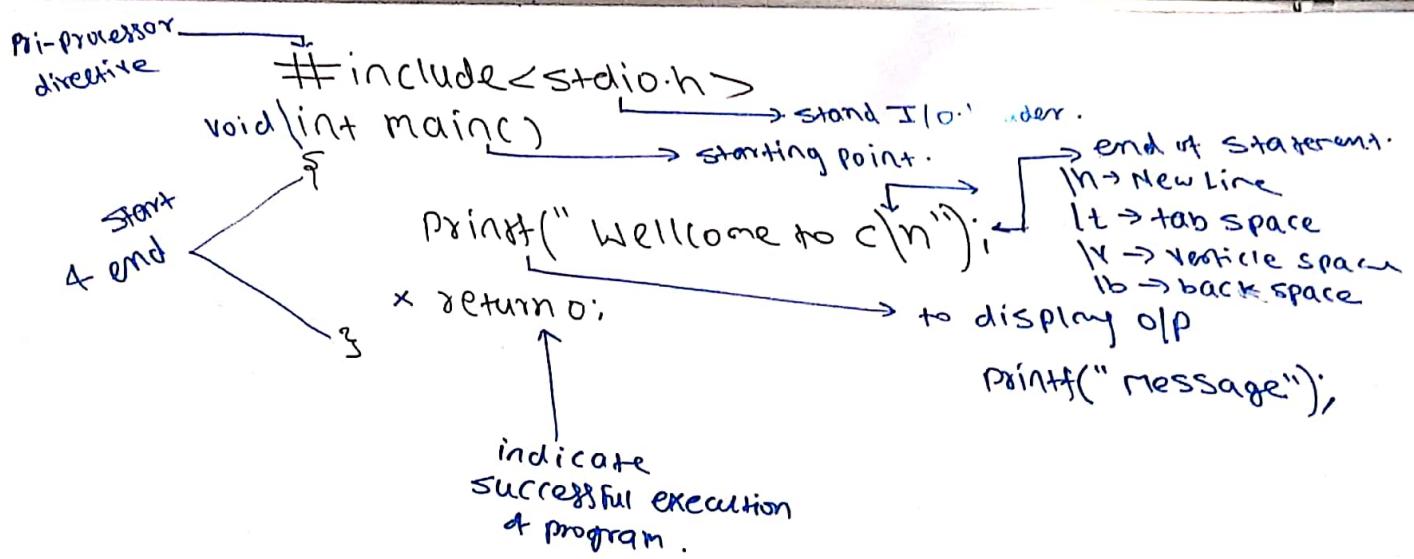


What is programming



Applications

- 1) scientific calcy.
- 2) virus & antivir.
- 3) Mobile & PC games.
- 4) Mozilla
- 5) Adobe



WAP to find area of circle

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

```
Void main()
```

```
{ int r;
```

```
float a;
```

```
Printf("Enter radius\n");
```

```
Scanf("%d", &r);
```

```
a = 3.14 * r * r;
```

```
Printf("Area = %F", a);
```

```
Getch();
```

↑
Get character.

$$a = \pi * r * r$$

%d int → 5, 2, 70 ...

%f float → 15.23 ...

%c char → a, A, b ...

DIF

Enter a radius

=

area = 154.00

WAP to accept basic salary and calculate gross salary including : % DA
40% HRA.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    float bs, gs, da, hra;
    printf("Enter basic salary\n");
    scanf("%f", &bs);
    da = 50/100 * bs;
    hra = 40/100 * bs;
    gs = bs + da + hra;
    printf("Gross Salary = %f", gs);
}
```

O/P

Enter basic salary

10000

Gross Salary = 19000.00

WAP to calculate roots of quadratic equation
by accepting coefficients from user.

```
ax2+bx+c=0 => x1, x2
#include <math.h>
#include <stdio.h>
#include <conio.h>
Void main()
{
    int a, b, c;
    float x1, x2;
    = 
$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

    printf("Enter coefficient of quad. equation\n");
    scanf("%d %d %d", &a, &b, &c);
    x1 = (-b + pow((b*b - 4*a*c), 0.5)) / 2*a;
    x2 = (-b - pow((b*b - 4*a*c), 0.5)) / 2*a;
    printf("Roots are x1=%f x2=%f", x1, x2);
    getch();
}
```

TOKENS

1) Character Set :-

Alphabet :- A-Z, a-z;

Digits :- 0-9

Symbol :- @, #, \$, %....

2) Keywords :- It is reserve words.

Total 32 keywords.

e.g:- int, float, if, break

3) Identifier :- Name given to memory, array, function, pointer.

Rules:

1) Combination of alphabet, digit & underscore.

2) Special symbols are not allowed except underscore.

3) Keywords are not allowed.

4) Blank space is not allowed.

5) Name cannot start with digit.

Variable & Constant :-

Variable :- place in computer memory to store data.

Syntax

datatype variable-name;

Constant :- Special variable whose data remain fixed.

Syntax

const datatype variable-name = value;

e.g:-

int x;

const float pi = 3.14;

Data type :- type of data.

Data-type

Primary

int

float

char

double

Secondary

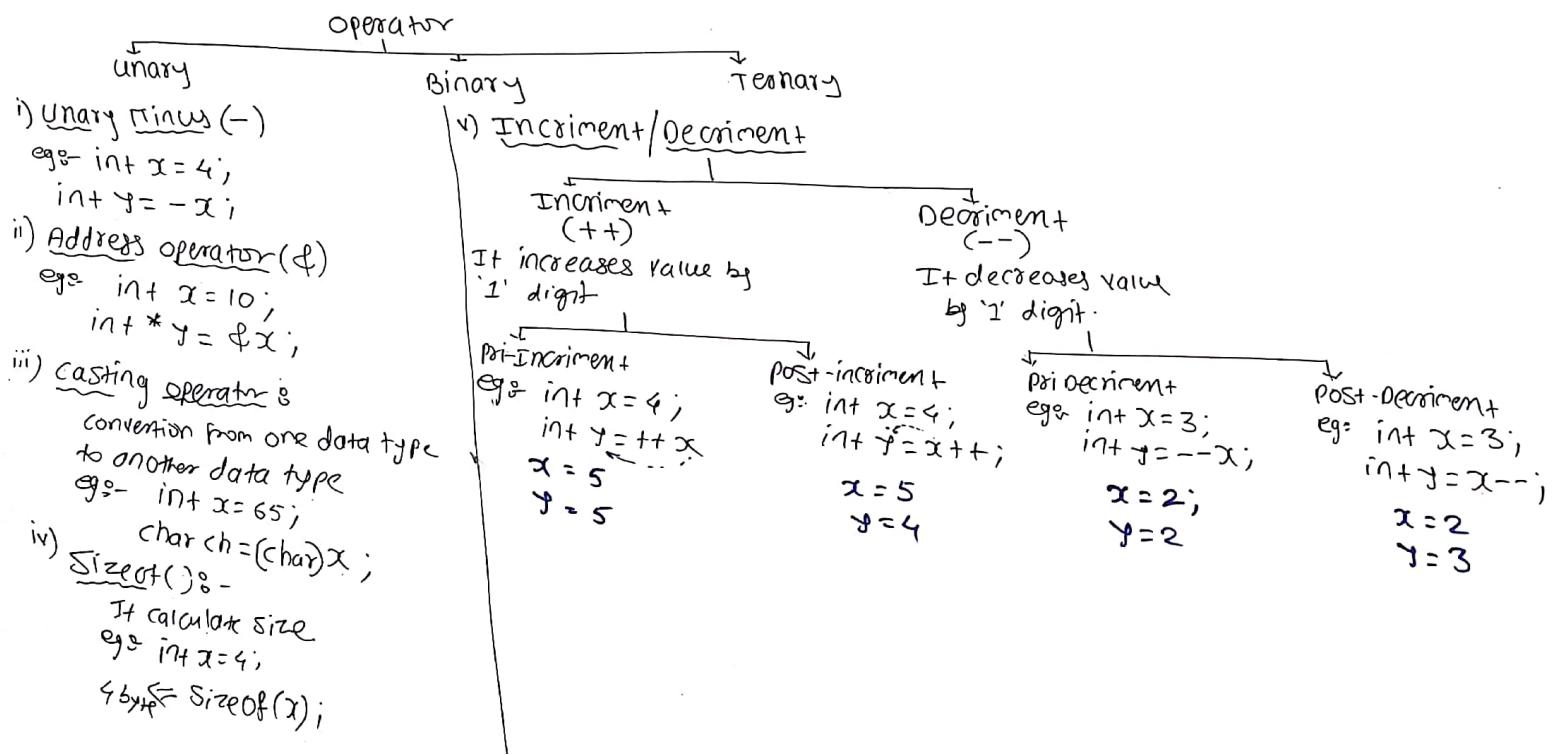
array

Pointer

Structure

Union

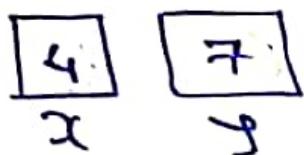
Operator :- Operator is symbol used to perform operations on operands.



```

#include<stdio.h>
#include<conio.h>
Void main()
{
    int x=4, y=9;
    int z;
    20  z = (x++) + (--y) + y;
    printf("Value = %d", z);
    16  z = (--x) + x + (y--);
    printf("Value = %d", z);
    getch();
}

```



O/P
Value = 20
Value = 16

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int a = 10, b, c;
    c = b = a;
    b -= a--;
    OR b = b - (a--);
    c -= --a;
    OR c = c - (--a);
    a -= (--a) - (a--);
    OR a = a - (-a - a--);
    printf("a=%d b=%d c=%d", a, b, c);
    getch();
}

```

6
0
2

a b c

Right then Left

O/P

a=6 b=0 c=2

```

#include<stdio.h>
#include<conio.h>           4) 
$$\begin{array}{r} 2 \\ 8 \\ \hline 8 \\ 0 \end{array}$$

Void main()
{
    int a=1, b=2, c=3, d=4.75, x;
    x = (++a) + (b++) * (++c) % d++;
    printf("%d %d %d %d %d", a, b, c, d, x);
    getch();
}

```

O/p

2 3 4 5 2

```

#include<stdio.h>
#include<conio.h>
Void main()
{
    int x = 1;
    printf("%d %d %d", x, 3x=1+2x+2, x<<2);
    x x<<2;
    printf("%d %d %d", ++x, 3x++, ++x);
    getch();
}

```

$x \leftarrow 1$ $x \leftarrow 1 + 2 = 3$ $x \ll 2 \Rightarrow 0100$
 $x \leftarrow 3$ $x \leftarrow 3 + 2 = 5$ $x \ll 2 \Rightarrow 1100$

O/P

3 3 4
5 4 4
//

Binary operator :-

1. Arithmetic operator :-

$+, -, *, /, \%$ are arithmetic operators.

'%' Mod, used to get remainder.

$$\begin{array}{r} 8421 \\ \hline 7 = 0111 \\ 9 = 1001 \\ 5 \quad 0101 \end{array}$$

2. Relational operator :-

$<, <=, >, >=, !=, ==$

Used to compare two operand.

3. Logical operator :-

1. AND ($\&$)

if (Condition1 $\&$ Condition2)

2. OR (||)

if (Condition1 || Condition2)

3. NOT (!)

if (! (marks >= 90))

4. Bitwise operator :-

Operator which works on bit of data.

i) Bitwise AND ($\&$)

$$\begin{array}{r} 5 \& 3 = (1) \\ 5 \Rightarrow 0101 \\ 3 \Rightarrow 0011 \\ \hline 1 \quad 0001 \end{array}$$

A	B	ANS
0	0	0
0	1	0
1	0	0
1	1	1

ii) Bitwise EX-OR

$$\begin{array}{r} 6 \Delta 4 = (2) \\ 6 \Rightarrow 0110 \\ 4 \Rightarrow 0100 \\ \hline 0010 \end{array}$$

A	B	ANS
0	0	0
0	1	1
1	0	1
1	1	0

v) Bitwise left shift (<<)

$$3 \ll 2 = (12)$$

3 times

$$3 \Rightarrow 0000000111$$

$$1^{\text{st}} \Rightarrow 0000001100$$

$$2^{\text{nd}} \Rightarrow 0000011000$$

vi) Bitwise Right Shift (>>) 13 >> 3 = (1)

A	B	ANS
0	0	0
0	1	0
1	0	1
1	1	1

iii) Bitwise NOT (\sim)

$$\begin{array}{r} \sim 7 = -(8) \\ 7 = 0111 \\ \hline 1000 \end{array}$$

Ternary Operator

It is also known as conditional operator.

Syntax

(condition) ? (Expr1) : (Expr2);

 true false

e.g - int n1=10;

int n2=15;

max = (n1 > n2) ? (n1) : (n2);

 10 15

WAP to find Largest of three number using conditional operator.

#include<stdio.h>

#include<conio.h>

void main()

{

 int n1, n2, n3, max;

 printf("Enter three number\n");

 scanf("%d %d %d", &n1, &n2, &n3);

 max = (n1 > n2) ? ((n1 > n3) ? n1 : n3) : ((n2 > n3) ? n2 : n3);

 printf("Largest Number = %d", max);

}

getch();

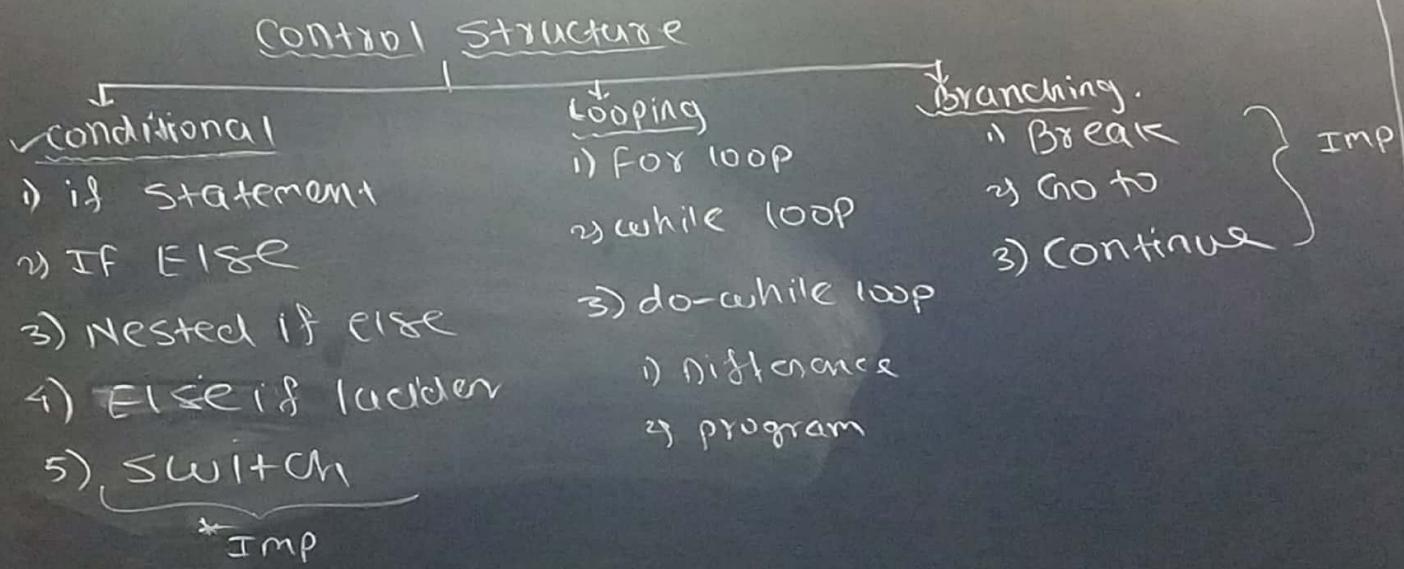
I/O function

↓
Formatted

1) printf();
2) scanf();

↓
Unformatted

- 1) getchar() : It accept single character
- 2) getch() : It accept single character does not echo it to screen.
- 3) getche() : It accept single character echo it to screen but does not wait for Enter Key to press.
- 4) putchar() : It display single character
- 5) gets() : It accept string (collection of character)
- 6) puts() : It display string .



i) IF Statement :-

Syntax
if (condition) ————— TRUE
 { statement; } ←

eg:-
if (marks >= 40) ————— ?
 {
 printf("you are pass\n");
 }
 ? 5

IF ELSE Statement

Syntax
if (condition) ————— T
 {
 statement 1; } ←
 else ————— F
 {
 statement 2; } ←
 }
 ? 38

eg:-
if (marks >= 40) ————— ?
 {
 printf("you are pass\n");
 }
 else ————— ?
 {
 printf("you are fail\n");
 }

WAP to check whether entered number is divisible by 10 or not

#include <stdio.h>

#include <conio.h>

void main()

{
 int n;
 printf("Enter a number\n");
 scanf("%d", &n);
 if (n % 10 == 0)
 s = 0;
 else
 printf("Number is divisible by 10");
 else
 printf("Number is not divisible by 10");
 getch();
}

$$\begin{array}{r} 2 \\ 10 \overline{) 20} \\ -20 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 3 \\ 10 \overline{) 30} \\ -30 \\ \hline 0 \end{array}$$

Nested if-Else

Syntax

if (condition-1)

{ if (condition-2)

{ Statement-1;

{ Statement-2;

} else

{ if (condition-3)

{ Statement-3;

} else

{ Statement-4;

}

}

Q9:- n1 n2 n3

if (n1 > n2)

{ if (n2 > n3)

{ printf("Largest number %d", n2);

{ printf("Largest number %d", n3);

} else

{ if (n2 > n3)

{ printf("Largest number %d", n2);

{ printf("Largest number %d", n3);

}

Else If Ladder

Syntax

```
if(condition-1)
    Statement -1;
else if(condition-2)
    Statement -2;
}
:
:
else
    Statement -n;
```

WAP to display class by accepting marks obtained

Marks	Class
75-100	Distinction
60-75	First class
40-60	Second class
<40	Fail

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int marks;
    printf("Enter marks\n");
    scanf("%d", &marks);
    if(marks >= 75 && marks <= 100)
        printf("Distinction\n");
    else if(marks >= 60 && marks < 75)
        printf("First class\n");
    else if(marks >= 40 && marks < 60)
        printf("Second class\n");
    else
        printf("Fail\n");
    getch();
}
```

O/P
Enter marks
45
Second class

Electric power distribution company charges following amount

Consumption unit	rate
0 - 200	0.5 per unit
201 - 400	Rs 100 + Rs 0.65 per unit for excess of 200
401 - 600	Rs 230 + Rs 0.85 per unit for excess of 400
601 - above	Rs 390 + Rs 1.00 per unit for excess of 600

Find total amount as per units consumed.

$$100 + 250 \times 0.65$$

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int unit;
    float rate;
    printf("Enter consumption unit\n");
    scanf("%d", &unit);
    if (unit <= 200)
        rate = 0.5 * unit;
    else if (unit > 200 && unit <= 400)
        rate = 100 + 0.65 * unit;
    else if (unit > 400 && unit <= 600)
        rate = 230 + (0.85 * unit);
    else
        rate = 390 + 1 * unit;
    printf("Bill amount is %f", rate);
}
```

O/P
Enter consumption unit
250

Bill amount is 380 Rs

```
else
    rate = 390 + 1 * unit;
printf("Bill amount is %f", rate);
getch();
```

Switch Statement

Syntax :-

```
switch (variable or expression)
{
    case label1 : statement1;
    break;
    case label2 : statement2;
    break;
    case label3 : statement3,
    break;
    .
    .
    default : statement-n;
}
```

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int m;
    printf("Enter month number\n");
    scanf("%d", &m);
    switch(m)
    {
        case 1 : printf("JAN");
        break;
        case 2 : printf("FEB");
        break;
        case 3 : printf("MAR");
        break;
        .
        .
    }
}
```

```
case 12 : printf("DEC");
break;
default : printf("Invalid month number");
break;
}
getch();
3
```

Switch Statement

Syntax:-

WAP to display number of days in a month using switch case.

switch
fall through.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int m;
    printf("Enter month number\n");
    scanf("%d", &m);
    switch(m)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12: printf("31 days");
                    break;
        case 4:
        case 6:
        case 9:
        case 11: printf("30 days");
                    break;
        case 2: printf("28/29 days");
                    break;
        default: printf("Invalid month number");
                    break;
    }
    getch();
}
```

```
case 4:
case 6:
case 9:
case 11: printf("30 days");
            break;
case 2: printf("28/29 days");
            break;
default: printf("Invalid month number");
            break;
```

Write a menu driven program to perform following operation
Addition/Subtraction/Multiplication/

Division

by accepting choice from user.

Q18

menu

1-ADD

2-SUB

3-MULT

4-DIV

Enter your choice

4

Enter two numbers

5

2

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n1,n2,ch;
    float ans;
    printf("Menu\n");
    printf(" 1-ADD\n 2-SUB\n 3-MULT\n 4-DIV\n");
    printf(" Enter your choice\n");
    scanf("%d", &ch);
    printf(" Enter two numbers\n");
    scanf("%d %d", &n1, &n2);
```

```
switch(ch)
{
    case 1:
        ans=n1+n2;
        printf("Addition = %f",ans);
        break;
    case 2:
        ans=n1-n2;
        printf("Subtraction = %f",ans);
        break;
    case 3:
        ans=n1*n2;
        printf("Multiplication = %f",ans);
        break;
    case 4:
        ans=(float)n1/n2;
        printf("Division = %f",ans);
        break;
    default:
        printf("Invalid choice\n");
}
getch();
```

Break Statement

Break is used to transfer the control to end of loop.

Syntax

```

break;
for (i = 1; i <= 10; i++)
{
    if (i == 5)
        break;
    printf("%d", i);
}
O/P
1 2 3 4

```

Continue

continue is used to transfer the control to beginning of loop.

Syntax

```

Continue;
for (i = 1; i <= 10; i++)
{
    if (i == 5)
        continue;
    printf("%d", i);
}
O/P
1 2 3 4 6 7 8 9 10

```

Goto

goto is used to transfer the control from one part of program to another part.

Syntax

```
goto (label);
```

```
label : 
```

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

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

```
void main()
```

```

int n;
accept();
printf("Enter a number");
scanf("%d", &n);
if (n < 0)
    goto accept;
printf("%d", n);
getch();

```

O/P

Enter a number

-5

Enter a number

-2

Enter a number

5

For Loop

Syntax

For(Initialization; condition; Increment)
 {
 Statement;

```
HELLO  

HELLO  

HELLO  

:  

FOR  

  HELLO  

  HELLO  

  HELLO  

  :  

  HELLO  

  HELLO  

  HELLO  

  :  

  HELLO
```

Initialization Condition Increment

for (i = 1; i <= 5; i++)
 {
 cout << "HELLO\n";
 }

WAP to find sum of 1 to N number

```
#include <stdio.h>
#include <conio.h>
void main()
{
  int n, sum=0, i;
  printf("Enter the value of N\n");
  scanf("%d", &n);
  for (i=1; i <= n; i++)
  {
    sum = sum + i;
  }
  printf("Sum of number = %d", sum);
  getch();
}
```

$i \uparrow 5$
 $1+2+3+4+5$

0	1	1	2	3	5	8
i	2	3	4	5			
n1	n2	ans		n			
		n2					

DIP

Enter no of terms

5

0	1	1	2	3	
n1	n2	ans			

WAP to display Fibonacci series upto number of terms

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

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

```
void main()
```

```
{ int n, n1, n2, ans, i;
```

```
printf("Enter no of terms\n");
scanf("%d", &n);
```

```
n1 = 0;
```

```
n2 = 1;
```

```
printf("%d\t %d\t ", n1, n2);
```

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

```
    ans = n1 + n2;
```

```
    printf("%d\t", ans);
```

```
    n1 = n2;
```

```
    n2 = ans;
```

```
}
```

```
getch();
```

WAP to check entered number
is prime number or not

O/P

ENTER a number

5

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, i, flag=0;
    printf("Enter a number\n");
    scanf("%d", &n);
    for(i=2; i<=n; i++)
    {
        if(n % i == 0)
            flag = 1;
        break;
    }
    if(flag==0)
        printf("Number is prime\n");
    else
        printf("Number is not prime\n");
}
```

While loop

Syntax

white (condition)
 |
 statement;

}

eg:-

int i = 1;
while (i <= 10) {
 printf("Hello\n");
 i++;}

false

010

Hello
Hello

WAP to reverse user entered number.

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

$$\begin{array}{r}
 153 \% 10 \Rightarrow 3 - rem \\
 153 / 10 \Rightarrow 15 \\
 15 \% 10 \Rightarrow 5 \\
 15 / 10 \Rightarrow 1 \\
 1 \% 10 \Rightarrow 1 \\
 1 / 10 \Rightarrow 0 \\
 \hline
 N = 0
 \end{array}$$

n	rem	rev
153	3	3
15	5	35
1	1	351
0		

15 1-
↓ ↓
15 1

Palindrome

WAP to check entered number is palindrome
or not

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

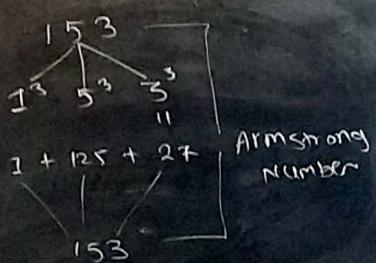
void main()

```
{ int n, rem, rev=0, temp;
printf("Enter a number\n");
scanf("%d", &n);
temp=n;
while(n!=0)
{
    rem=n%10;
    rev=rev*10+rem;
    n=n/10;
}
if(temp==rev)
    printf("It is palindrome\n");
else
    printf("It is not palindrome\n");
getch();}
```

$$\begin{array}{r} 153 \% 10 \Rightarrow 3 - \text{rem} \\ 153 / 10 \Rightarrow 15 \\ 15 \% 10 \Rightarrow 5 \\ 15 / 10 \Rightarrow 1 \\ 1 \% 10 \Rightarrow 1 \\ 1 / 10 \Rightarrow 0 \\ \hline \boxed{n = 0} \end{array}$$

n	rem	rev
15	1	1
15	5	15
1	1	151
0		

$$10 \overline{) \underline{1}}^{\underline{0}}$$



WAP to check entered number is Armstrong Number.

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

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

```
void main()
```

```
{ int n, rem, sum=0, temp;
printf("Enter a number\n"),
```

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

```
temp=n;
```

```
while(n!=0) . . .
```

```
rem=n%10;
```

```
sum=sum+rem*rem*rem;
```

```
n=n/10;
```

```
if(temp==sum)
```

```
printf("It is Armstrong\n");
```

```
else
```

```
printf("It is not Armstrong\n");
```

```
} getch();
```

$$153 \% 10 \Rightarrow 3 - \text{rem}$$

$$153 / 10 \Rightarrow 15$$

$$15 \% 10 \Rightarrow 5$$

$$15 / 10 \Rightarrow 1$$

$$1 \% 10 \Rightarrow 1$$

$$1 / 10 \Rightarrow 0$$

$$N = 05$$

n	rem	rem
151	1	1
15	5	15
1	1	151
0		

WAP to display following QP

1 2 3
4 5 6
7 8 9 10

*
**|
*** *|
*** * *

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

printf("%d", i))

1
1 2
1 2 3
1 2 3 4

printf("%d", j))

||

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

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

```
void main()
```

```
{ int i, j, k = 1;
```

```
for (i = 1; i <= 4; i++) {
```

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

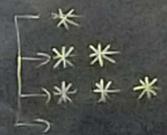
```
        printf("%d", k),
```

```
        k++;
```

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

```
getch();
```

```
}
```



WAP to display following dp

*
* *
* * *
* * * *

char	ASCII	
A	65	A
B	66	B B
C	67	C C C
D	68	D D D D

1 65
2 66 66
3 67 67 67

A
A B
A B C
A B C D

printf("%c", (char)(64+i));

#include < stdio.h >

#include < conio.h >

void main()

```
{ int i, j;
for (i = 1; i <= 4; i++)
{
    for (j = 1; j <= i; j++)
        printf("%c", (char)(64+i));
    printf("\n");
}
```

getch();

WAP to display following app

	*	*	*	*
1	*	*	*	*
2	-	*	*	*
3	-	*	*	*
4	*	*	*	*

$$n-i$$

$$= 4-1 = 3$$

$$= 4-2 = 2$$

$$= 4-3 = 1$$

$$= 4-4 = 0$$

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

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

```
void main()
```

```
{
```

```
int i, j;
```

```
for (i = 1; i <= 4; i++)
```

```
{
```

```
    for (j = 1; j <= 4 - i; j++)
```

```
{
```

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

```
}
```

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

```
{
```

```
    printf("*");
```

```
}
```

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

```
}
```

```
getch();
```

```
}
```

O/P

```
-----*
```

```
-----*
```

```
-----*
```

```
-----*
```

WAP to display following dp

* * * *
* * * *
* * * *
* * * *
- - - 1 ← -
1 2 3
1 2 3 ←

A
A B
A B C
A B C D
printf("%c", (char)(64+i));

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

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

```
void main()
```

```
{ int i, j;
```

```
for (i = 1; i <= 4; i++)
```

```
{ for (j = 1; j <= 4 - i; j++)
```

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

```
}
```

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

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

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

```
 }
```

O/P

→ ----- * *

WAP to display following app

```
1 - - - * | 1 - - - 2-1  
2 - - * * * | 3-1-2  
3 - * * * * * |  
4 * * * * * |  
- - - 1 <-1  
- 1 2  
1 2 3  
1 2 3 4
```

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

```
printf("%c", (char)(64+i));
```

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
    int i, j;  
    for(i = 1; i <= 4; i++)  
    {  
        for(j = 1; j <= 4-i; j++)  
            printf(" ");  
        for(j = 1; j <= i; j++)  
            printf("*");  
        for(j = 1; j <= i-1; j++)  
            printf("*");  
        printf("\n");  
    }  
    getch();
```

O/P

```
-----* * *  
-----* * * *  
-----* * * * *
```

WAP to display following O/P

- - -
 1 2 1 2 1
1 2 3 4 A B C

 A
 A B
 A B C
A B C D

printf("%c", (char)(64+i))

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

```
void main()
```

```
{ int i, j;  
  for (j = 1; j <= i; j++)
```

O/P

```
  { for (j = 1; j <= i-1; j++)
```

- - - * * *
-> * * * * *

```
    printf(" ");
```

```
  }  
  for (j = 1; j <= i; j++)
```

```
    printf("%d", j);
```

```
  for (j = 1; j <= i-1; j++)
```

```
    printf("%c", (char)(64+j));
```

```
  printf("\n");
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

WAP to display following QP

```

    1
   - - *
  - * * *
 - * * * *

```

```

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

```

O/P

```

    - - -
   - * * *
  - * * * *

```

WAP to display following QP

odd [1 1
2 2 1
3 1 2 3
4 4 3 2 1]
Even
void main()
{ int i, j;
for(i = 1; i <= 4; i++)
{ if(i % 2 != 0)
 for(j = 1; j <= i; j++)
 printf("%d", j);
 else
 for(j = i; j >= 1; j--)
 printf("%d", j);
 printf("\n");
}
getchar();

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

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

```
void main()
```

```
{ int i, j;  
for(i = 1; i <= 4; i++)  
{ for(j = 1; j <= 4 - i; j++)  
    printf(" ");  
    for(j = i; j <= 4 - i + 1; j++)  
        printf("%c", (char)(64 + j));  
    printf("\n");  
}
```

Q/P

```
-----*  
-----* * * *  
-----*
```

WAP to display following pattern

```
----- A 1  
      C B 2  
      F E D 3  
      J I H G 4  
O N M L K S
```

```
#include < stdio.h >  
#include < conio.h >  
void main()  
{  
    int i, j, k = 0, m;  
  
    for (i = 1; i <= 5; i++)  
    {  
        k = 2 * i - 1;  
        m = k;  
        for (j = 1; j <= 5 - i; j++)  
        {  
            printf(" ");  
        }  
        for (j = 1; j <= k; j++)  
        {  
            printf("%c", (char)(65 + m));  
            m--;  
        }  
        printf("\n");  
    }  
}
```

----- C B A

Function :-

function is subprogram used to perform particular task.

function contains

- ✓ 1. Function definition
- ✓ 2. Calling function
- 3. Function prototype

Function Definition

Syntax

```
return-type   function-name (Arguments)
{           }
    Statement;
}
```

Call to Function

function-name (Parameters),

Function Declaration / Prototype

WAP to perform addition of two numbers

```
#include<stdio.h>
#include<conio.h>
int Add(int, int); // Prototype
```

```
int n1, n2, ans,
printf("Enter two numbers\n");
scanf("%d %d", &n1, &n2);
ans = Add(n1, n2); // Call to function
printf("Addition = %d", ans);
getch();
```

}

```
int Add(int x, int y)
{
    int z;
    z = x + y;
    return(z);
}
```

O/P
Enter two numbers
5
7
Addition = 12

FUNCTIONS

function is subprogram used to perform particular task
#include <stdio.h>

1) function definition

2) calling function

3) function prototype

func. definition

states

return-type function-name(Arguments)

Statement;

call to function

function-name(Parameters),

func. declaration/prototype

#include <stdio.h>
int fact(int n);
void main()
{
 int n, ans;
 printf("Enter a number\n");
 scanf("%d", &n);
 ans = fact(n);
 printf("Factorial = %d", ans);
}

int fact(int n)

{
 int i, f=1;
 for(i=2; i<=n; i--)
 f = f * i;
 return(f);
}

Q1

Enter a number

4

factorial = 24.

Date _____

WAP to find value of n_p_r using function.

```

#include<stdio.h>
#include<conio.h>
int fact(int x);
Void main()
{
    int n, r, ans;
    printf("Enter the value of n & r\n");
    scanf("%d %d", &n, &r);
    ans = fact(n) / fact(r);
    printf("npr = %d", ans);
    getch();
}

```

$$n_p_r = \frac{n!}{(n-r)!}$$

$$5_p_2 = \frac{5!}{3!} = 20$$

```

int fact(int x)
{
    int i, f=1;
    for(i=x; i>=1; i--)
    {
        f = f * i;
    }
    return(f);
}

O/P
Enter a number
factorial = 24

```

$5 = 5 * 4 * 3 * 2 * 1$
 $4 = 4 * 3 * 2 * 1$

```

#include<std.h>
#include<conio.h>
int fact(int x);
void main()
{
    int i, j, n, a;
    printf("Enter no of lines\n");
    scanf("%d", &n),
    for(i=0; i<=n; i++)
    {
        for(j=0; j<=n-i; j++)
            printf("-");
        printf("\n");
    }
    a = fact(i)/(fact(j)*fact(i-j));
    printf("%d", a),
    printf("\n");
}

```

```
O/P  
Enter a number  
4  
Factorial = 24
```

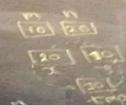
Call by Value & Call by Reference.

1) Call by Value :-

It passes copy of value to function.

```

Swapping
#include<stdio.h>
#include<conio.h>
void swap(int i,int j)
{
    int m,n,
    printf("Enter two numbers\n"),
    scanf("%d %d",&m,&n),
    printf("Value before swap\n"),
    printf("m = %d n = %d",m,n),
    swap(m,n),
    printf("Value after swap\n"),
    printf("m = %d n = %d",m,n),
    getch(),
}
  
```



void swap(int i,int j)
{
 int temp;
 temp = i;
 i = j;
 j = temp;

3

O/P
ENTER two numbers
10
20
Value before swap
m=10 n=20
Value after swap
m=10 n=20

call by value & call by reference

2. call by reference
It passes address (reference)
of value to function
so any changes made by func
will affect original value also

Pointers
Used to store address of
another variable

eg:-
 $\text{int } *p;$ $*p$ [10] [1010]

$\text{int } x = 10;$ 10 [1010] P

$p = \&x;$

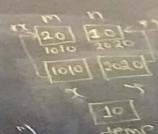
$\text{printf(" \%d", } x);$ 10

$\text{printf(" \%d", } *p);$ 10

$\text{printf(" \%u", } p);$ 1010

Swapping

```
#include<stdio.h>
#include<conio.h>
void swap(int*, int*);
void main()
{
    int m, n,
    printf("Enter two numbers\n"),
    scanf("%d %d", &m, &n),
    printf("Value before swap\n"),
    printf("m = %d n = %d", m, n),
    swap(&m, &n),
    printf("Value after swap\n"),
    printf("m = %d n = %d", m, n),
    getch(),
}
```



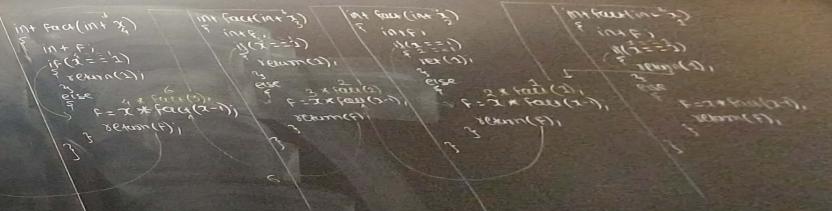
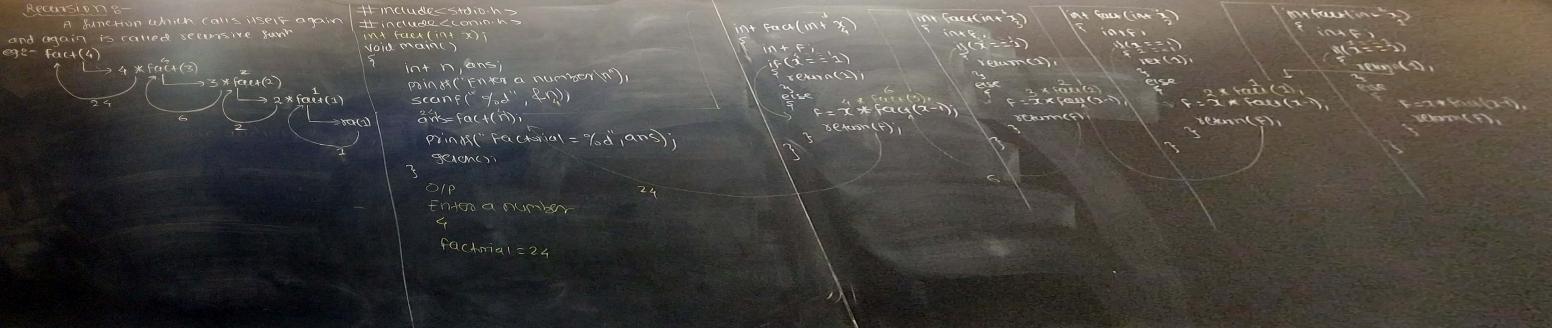
void swap(int* x, int* y)

```
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
```

O/P

Enter two numbers
10
20

Value before swap
m = 10 n = 20
Value after Swap
m = 20 n = 10

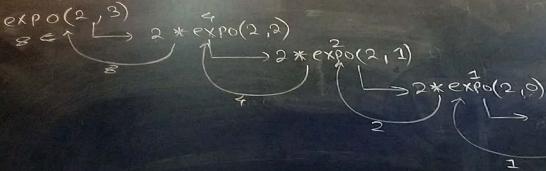


$$\text{ans} = \text{fact}(n) / (\text{fact}(x) * \text{fact}(n-x));$$

Write a program to find value of y where $y = x^n$ using recursion.

$$x=2, n=3$$

$$2^3 \Rightarrow 8$$



```
int expo(int x){  
    #include<csio.h>  
    #include<conio.h>  
    void main()  
    {  
        int x,n,y;  
        printf("Enter the value of x & n\n");  
        scanf("%d %d", &x, &n);  
        y = expo(x, n);  
        printf("exponential = %d", y);  
        getch();  
    }  
}
```

O/P
Enter the Value of x & n
2
3
Exponential = 8

```
int expo(int x, int n)  
{  
    int y;  
    if(n == 0)  
    {  
        y = 1;  
        return(y);  
    }  
    else  
    {  
        y = x * expo(x, n-1);  
    }  
    return(y);  
}
```

ARRAY :- array is used to store collection of values of similar data type

Syntax

data-type array-name[size];

e.g -

int a[5];

a	[10	20]
index	0	1	2	3	4			

Starting index = 0

Last index = n-1

Initialize array

1) int a[] = {10, 20, 30, 40, 50};

2) int a[5];

a[0] = 10,

a[1] = 20,

WAP to find largest element present in array

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, a[20], i, max;
    printf("Enter no of elements\n");
    scanf("%d", &n);
    printf("Enter the values\n");
    for(i=0; i<n; i++)
    {
        scanf("%d", &a[i]);
    }
}
```

5	1	7	9	3
0	1	2	3	4

max = a[0];

if(a[4] > max)

 max = a[3];

max = a[0];

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

{

 if(a[i] > max)

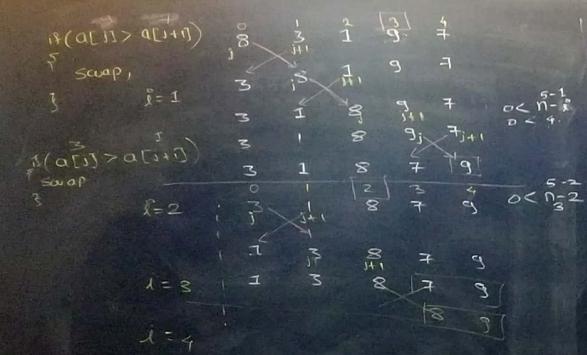
 max = a[i];

}

printf("largest number = %d", max);

getch();

}



WAP to sort numbers in ascending order
 using bubble sort

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, a[20], i, temp, j,
        printf("Enter no of elements(n):"),
        scanf("%d", &n),
        printf("Enter the values(n):"),
        for(i=0; i<n; i++)
            scanf("%d", &a[i]);
    for(i=0; i<n-1; i++)
        for(j=0; j<n-i-1; j++)
            if(a[j] > a[j+1])
                {
                    temp = a[j];
                    a[j] = a[j+1];
                    a[j+1] = temp;
                }
    printf("sorted list is\n"),
    for(i=0; i<n; i++)
        printf("%d\t", a[i]);
    getch();
}
```

$a[1\ 3\ 4\ 1\ 8\ 7]$
 $\downarrow\downarrow\downarrow\downarrow\downarrow\downarrow$
 $i=0$
 $i=1$
 $i=2$
 $i=3$
 $i=4$

WAP to search number present in array
using iteration

```
#include<stdio.h>
#include<conio.h>
int search(int arr[], int key, int n);
Void main()
{
    int n, arr[100], key;
    printf("Enter no of elements\n");
    scanf("%d", &n);
    printf("Enter the values\n");
    for(i=0; i<n; i++)
        scanf("%d", &arr[i]);
    printf("Enter number to search\n");
    scanf("%d", &key);
    if(i == -1)
        printf("Not found\n");
    else
        printf("Element found at %d", i+1);
}
```

int search(int arr[], int key, int n)

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

{
 if(arr[i] == key)
 return(i);
 }
}

return(-1);

int search(int arr[], int key, int n)

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

{
 if(arr[i] == key)
 return(i);
 }
}

return(-1);

Two Dimensional arrays

```

SYNTAX
double-type array-name[rowsize][colszie];
example: a[3][3]
          int a[3][3];
          a[0][0] = 10; a[0][1] = 20; a[0][2] = 30;
          a[1][0] = 40; a[1][1] = 50; a[1][2] = 60;
          a[2][0] = 70; a[2][1] = 80; a[2][2] = 90;
          cout << a[0][0] << a[0][1] << a[0][2] << endl;
          cout << a[1][0] << a[1][1] << a[1][2] << endl;
          cout << a[2][0] << a[2][1] << a[2][2] << endl;

```

```

WAP for addition two matrix of size m x n
#include < stdlib.h>
#include <conio.h>
void main()
{
    int m, n, a[10][10], b[10][10], c[10][10], i, j;
    printf("Enter no of rows & columns");
    scanf("%d %d", &m, &n);
    printf("Enter the value of Matrix-A");
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }
    printf("Enter the value of Matrix-B");
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            scanf("%d", &b[i][j]);
        }
    }
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            c[i][j] = a[i][j] + b[i][j];
        }
    }
    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            cout << c[i][j] << " ";
        }
        cout << endl;
    }
}

```

```

printf("Enter the value of matrix A");
for (i = 0; i < m; i++)
{
    for (j = 0; j < n; j++)
    {
        scanf("%d", &a[i][j]);
    }
}
printf("Enter the value of matrix B");
for (i = 0; i < m; i++)
{
    for (j = 0; j < n; j++)
    {
        scanf("%d", &b[i][j]);
    }
}
for (i = 0; i < m; i++)
{
    for (j = 0; j < n; j++)
    {
        c[i][j] = a[i][j] + b[i][j];
    }
}
for (i = 0; i < m; i++)
{
    for (j = 0; j < n; j++)
    {
        cout << c[i][j] << " ";
    }
    cout << endl;
}

```

```

printf("Addition of two matrix");
for (i = 0; i < m; i++)
{
    for (j = 0; j < n; j++)
    {
        cout << a[i][j] << " ";
    }
    cout << endl;
}
cout << endl;
cout << "Addition of two matrix";
for (i = 0; i < m; i++)
{
    for (j = 0; j < n; j++)
    {
        cout << b[i][j] << " ";
    }
    cout << endl;
}
cout << endl;
cout << "Addition of two matrix";
for (i = 0; i < m; i++)
{
    for (j = 0; j < n; j++)
    {
        cout << c[i][j] << " ";
    }
    cout << endl;
}
cout << endl;

```

WAP for multiplication of two matrix.

$$\begin{array}{c}
 \text{A} \\
 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \\
 \times
 \end{array}
 \begin{array}{c}
 \text{B} \\
 \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \end{bmatrix} \\
 = \begin{bmatrix} 1 & 2 & 4 & 6 \end{bmatrix}
 \end{array}$$

$\sum_{j=1}^3 \sum_{k=1}^2 A_{ij} B_{jk} = C_{ik}$

$$\begin{aligned}
 C[i][j] &= C[i][j] + a[i][k] * b[k][j] \\
 &= (\text{Ans}) + a[i][j]
 \end{aligned}$$

$$\text{Ans} = \text{Ans} + a[i][j]$$

WAP for addition of two matrix of size MxN

```

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

void main()
{
    int i, j, m1[3][3], m2[3][3], m3[3][3];
    printf("Enter no of rows & columns of matrix A");
    scanf("%d %d", &m1[0][0], &m1[0][1]);
    printf("Enter the value of matrix A");
    for(i=0; i<m1[0][0]; i++)
        for(j=0; j<m1[0][1]; j++)
            scanf("%d", &m1[i][j]);
    printf("Enter no of rows & columns of matrix B");
    scanf("%d %d", &m2[0][0], &m2[0][1]);
    printf("Enter the value of matrix B");
    for(i=0; i<m2[0][0]; i++)
        for(j=0; j<m2[0][1]; j++)
            scanf("%d", &m2[i][j]);
    for(i=0; i<m1[0][0]; i++)
        for(j=0; j<m2[0][1]; j++)
            m3[i][j] = m1[i][j] + m2[i][j];
    for(i=0; i<m1[0][0]; i++)
        for(j=0; j<m1[0][1]; j++)
            printf("%d ", m3[i][j]);
}

```

Print("Enter the value of matrix A");

For(i=0; i<m1[0][0]; i++)

{ For(j=0; j<m1[0][1]; j++)

scanf("%d", &m1[i][j]);

Print("Enter the value of matrix B");

For(i=0; i<m2[0][0]; i++)

{ For(j=0; j<m2[0][1]; j++)

scanf("%d", &m2[i][j]);

Print("Addition of two matrix");

For(i=0; i<m1[0][0]; i++)

{ For(j=0; j<m2[0][1]; j++)

m3[i][j] = m1[i][j] + m2[i][j];

For(i=0; i<m1[0][0]; i++)

{ For(j=0; j<m1[0][1]; j++)

printf("%d ", m3[i][j]);

Print("Addition of two matrix");

For(i=0; i<m1[0][0]; i++)

{ For(j=0; j<m1[0][1]; j++)

scanf("%d", &m1[i][j]);

Print("Enter the value of matrix B");

For(i=0; i<m2[0][0]; i++)

{ For(j=0; j<m2[0][1]; j++)

scanf("%d", &m2[i][j]);

Print("Addition of two matrix");

For(i=0; i<m1[0][0]; i++)

{ For(j=0; j<m2[0][1]; j++)

m3[i][j] = m1[i][j] + m2[i][j];

For(i=0; i<m1[0][0]; i++)

{ For(j=0; j<m1[0][1]; j++)

printf("%d ", m3[i][j]);


```

WAP for addition two matrix of size m x n
#include < stdio.h>
#include <conio.h>

void main()
{
    int i, j, a[10][10], b[10][10], c[10][10], r1, c1, r2, c2;
    printf("Enter no of rows & columns of matrix A\n");
    scanf("%d %d", &r1, &c1);
    printf("Enter no of rows & columns of matrix B\n");
    scanf("%d %d", &r2, &c2);
    if(r1 != r2 || c1 != c2)
    {
        printf("Matrix multiplication not possible");
        exit(0);
    }
    else
    {
        for(i = 0; i < r1; i++)
        {
            for(j = 0; j < c1; j++)
            {
                printf("Enter the value of matrix A[%d][%d]\n", i, j);
                scanf("%d", &a[i][j]);
            }
        }
        for(i = 0; i < r2; i++)
        {
            for(j = 0; j < c2; j++)
            {
                printf("Enter the value of matrix B[%d][%d]\n", i, j);
                scanf("%d", &b[i][j]);
            }
        }
        for(i = 0; i < r1; i++)
        {
            for(j = 0; j < c2; j++)
            {
                c[i][j] = 0;
                for(k = 0; k < c1; k++)
                {
                    c[i][j] = c[i][j] + a[i][k] * b[k][j];
                }
            }
        }
        printf("Resultant matrix C\n");
        for(i = 0; i < r1; i++)
        {
            for(j = 0; j < c2; j++)
            {
                printf("%d ", c[i][j]);
            }
            printf("\n");
        }
    }
}

```

```

printf("Enter no of rows & columns of matrix A\n");
for(i=0; i<r1; i++)
{
    for(j=0; j<c1; j++)
    {
        scanf("%d", &a[i][j]);
    }
}
printf("Enter no of rows & columns of matrix B\n");
for(i=0; i<r2; i++)
{
    for(j=0; j<c2; j++)
    {
        scanf("%d", &b[i][j]);
    }
}
for(i=0; i<r1; i++)
{
    for(j=0; j<c2; j++)
    {
        c[i][j] = 0;
        for(k=0; k<c1; k++)
        {
            c[i][j] = c[i][j] + a[i][k] * b[k][j];
        }
    }
}

```

```

printf("Enter no of rows & columns of matrix A\n");
for(i=0; i<r1; i++)
{
    for(j=0; j<c1; j++)
    {
        scanf("%d", &a[i][j]);
    }
}
printf("Enter no of rows & columns of matrix B\n");
for(i=0; i<r2; i++)
{
    for(j=0; j<c2; j++)
    {
        scanf("%d", &b[i][j]);
    }
}
for(i=0; i<r1; i++)
{
    for(j=0; j<c2; j++)
    {
        c[i][j] = 0;
        for(k=0; k<c1; k++)
        {
            c[i][j] = c[i][j] + a[i][k] * b[k][j];
        }
    }
}

```

Print("Multiplication of two matrix(n), ");

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

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

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

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

 }

 cout << endl;

}

Write a program to multiply two matrix by using following function
1) accept matrix 2) matmultiplication 3) display

```
if(c1 == c2)
    matmultiplication(a, b, c, r1, c2, c1);
else
    printf("Multiplication not possible\n");
    return;
```

```
WAP for addition two matrix of size m x n
#include <stdio.h>
#include <conio.h>

void main()
{
    int i, j, k;
    int r1, c1, r2, c2;
    printf("Enter no of rows & columns of Matrix-A\n");
    scanf("%d %d", &r1, &c1);
    printf("Enter the value of Matrix-A\n");
    accept(a, r1, c1);
    printf("Enter no of rows & columns of Matrix-B\n");
    scanf("%d %d", &r2, &c2);
    printf("Enter the value of Matrix B\n");
    accept(b, r2, c2);
```

```
void accept(int x[10][10], int m, int n)
{
    for(i=0; i<m; i++)
        for(j=0; j<n; j++)
            scanf("%d", &x[i][j]);
}
```

```
void matmultiplication(int a[10][10], int b[10][10],
int r1, int c1, int r2, int c2, int r3, int c3)
{
    for(i=0; i<r3; i++)
        for(j=0; j<c3; j++)
            C[i][j] = 0;
    for(i=0; i<r1; i++)
        for(j=0; j<c2; j++)
            for(k=0; k<c1; k++)
                C[i][j] = C[i][j] + a[i][k] * b[k][j];
}
```

```
void display(int x[10][10], int m, int n)
{
    for(i=0; i<m; i++)
        for(j=0; j<n; j++)
            printf("%d", x[i][j]);
    printf("\n");
}
```

String :-

It is array of character.

Syntax

```
char string-name [size];
```

Eg:-

```
char str[10];
```

str

H	e	l	l	o	\0	i	i	l	i
0	1	2	3	4	5	6	7	8	9

```
char str[] = "Hello";
```

'\0' is used to terminate string.

String Handling Function

1. strlen() :- This fun is used to find length of a string.

Syntax

```
int strlen(string);
```

Eg:-

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
Void main()
{
    char str[20];
    int len;
    printf("Enter a String\n");
    gets(str);
    len = strlen(str);
    printf("Length of a String %d", len);
}
```

WAP to Find the length of String without using "string.h".

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

{

```
    char str[20];
```

```
    int len=0, i=0;
```

```
    printf("Enter a string\n");
```

```
    gets(str);
```

```
    while (str[i] != '\0')
```

```
        len++;
```

```
        i++;
```

```
}
```

```
    printf("length of a string = %d", len);
}
```

str	0	1	2	3	4	5
	H	e	l	l	o	\0

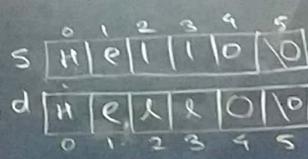
len
5

len

WAP to Find to copy using "String.h".

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char s[20], d[20], d[5];
    int len=0, i=0;
    printf("Enter a string\n");
    gets(s);
    while(s[i] != '\0')
    {
        d[i] = s[i];
        i++;
    }
    d[i] = '\0';
    printf("String after copy = %s", d);
    getch();
}
```

String without



String Handling Function

1. `strcpy()`: This function is used to copy one string into another string.

Syntax

```
strcpy(dest_string, source_string);
```

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

```
{ char s[20], d[20];
printf("Enter a string\n");
gets(s);
```

```
strcpy(d, s);
```

```
printf("String after copy %s", d);
getch();
```

WAP to Find to copy
using "string.h".

```
#include<stdio.h>
#include<conio.h>
void main()
{
    printf("String
after concat %s",
    );
    getch();
}

char s[20], d[20], d [H | e | l | l | o | \0]
int i=0, j=0;
printf("Enter a first String\n"),
gets(s);
printf("Enter a second String\n"),
gets(d);
while(s[i]!='\0')
{
    i++;
}
while (d[j]!='\0')
{
    s[i] = d[j];
    i++;
    j++;
}
s[i] = '\0';
```

String without

String Handling function

3. strcat(): This fun is used to concat (join)
two String together.

Syntax

strcat (String1, String2);

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char s[20], d[20];
    printf("Enter first string\n");
    gets(s);
    printf("Enter second string\n");
    gets(d);
    strcat(s, d);
    printf("String after concat = %s", s);
    getch();
}
```

WAP to check entered string is palindrome or not

```
#include<string.h>
#include<stdio.h>
#include<conio.h>
void main()
{
    char str[20];
    int len, i, flag = 0;
    printf("Enter a String\n");
    gets(str);
    len = strlen(str),
    for(i = 0; i < (len/2); i++)
    {
        if(str[i] != str[len - 1 - i])
        {
            flag = 1;
            break;
        }
    }
}
```



```
if(flag == 0)
{
    printf("String is palindrome\n");
}
else
{
    printf("String is not palindrome\n");
}
getch();
}
```

```
if(i == 0)
{
    printf("Strings are equal\n");
}
else
{
    printf("Strings are not equal\n");
}
getch();
}
```

Strcmp() :- This functn is used to compare two string

Syntax:-

```
int strcmp(String1, String2);
```

```
#include <stdio.h>
#include <iostream>
#include <string.h>
void main()
```

```
{
```

```
char s[20], d[20];
```

```
printf("enter first string\n");
gets(s);
```

```
printf("enter second string\n");
gets(d);
```

```
i = strcmp(s, d)
```

```
if (i == 0)
```

```
{
```

```
printf("Strings are equal\n")
```

```
else
```

```
{
```

```
printf("Strings are not equal\n")
```

```
{
```

```
getch();
```

```
}
```

Structure :-
Structure is user defined datatype
which can store mix type of data.

Example

```
struct struct.name  
{  
    Datatype Variable1,  
    Datatype Variable2,  
    ...  
};
```

3)

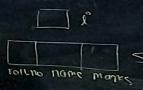
WAP to create Structure Student
having data member rollno, name & marks.
accept the data for one student & display.

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

```
void main()  
{  
    int i;  
    struct Student s;
```

```
    int rollno;  
    char name[20];  
    float marks;
```

```
}  
};
```



```
printf("Enter roll number\n");  
scanf("%d", &s.rollno);  
printf("Enter student Name\n");  
gets(s.name);  
printf("Enter marks\n");  
scanf("%f", &s.marks);  
printf("Student details are\n");  
printf("Roll No = %d\n Name = %s\n Marks = %f", s.rollno, s.name, s.marks);  
getch();
```

WAP to create nested structure to store details of n student with name, rollno & marks of phy, chem, maths. display the details of student in tabular form as Rollno Name Total Marks

```
#include<stdio.h>
#include<conio.h>
void main()
{
    struct Student
    {
        int rollno;
        char name[20];
        struct
        {
            int phy;
            int chem;
            int maths;
        }marks;
        int total;
    };
    struct Student S[10];
    int n;
    printf("Enter no of student\n");
    scanf("%d", &n);
    for(i=0; i<n; i++)
    {
        printf("Enter roll no\n");
        scanf("%d", &S[i].rollno);
        printf("Enter name\n");
        gets(S[i].name);
        printf("Enter marks of phy, chem, maths\n");
        scanf("%d %d %d", &S[i].marks.phy, &S[i].marks.chem, &S[i].marks.maths);
        S[i].total = S[i].marks.phy + S[i].marks.chem + S[i].marks.maths;
    }
}
```

Rollno	Name	Physics	Chem	Maths	Total
1	John	85	78	80	243
2	Jane	72	80	75	227
3	Mike	88	85	82	255
4	Alice	75	70	78	223
5	Bob	80	75	70	225

```
printf("Rollno Name Total\n");
for(i=0; i<n; i++)
{
    printf("%d %s %d\n", S[i].rollno, S[i].name, S[i].total);
}
getchar();
```

WAP to Create Structure employee
having data member empid, ename,
salary accept details of 10 employee
of display details of those employee
whose salary greater than 50000

equal to

Sat 8 2 PM

File of pointer

```
#include<stdio.h>
#include<conio.h>
struct employee
{
    int empid;
    char ename[10];
    float sal;
};

void main()
{
    struct employee e[10];
    int i;
    for(i=0; i<10; i++)
    {
        printf("Enter emp id\n");
        scanf("%d", &e[i].empid);
        printf("Enter name\n");
        gets(e[i].ename);
        printf("Enter salary\n");
        scanf("%f", &e[i].sal);
    }
}
```

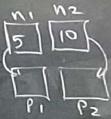
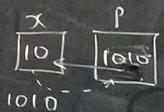
```
printf("Emp id Emp Name Emp Salary\n");
for(i=0; i<10; i++)
{
    if(e[i].sal >= 50000)
        printf("%d %s %f\n", e[i].empid, e[i].name, e[i].sal);
}
getch();
```

Pointers :- Pointer is variable which is used to store the address of another variable

Syntax

data-type *pointer-name;

e.g:-
int *p;
char *c;
float *ptr;



printf("%d", x); // 10
printf("%d", *p); // 10

1) Reference operator:

$\text{int } i \text{ is also known as address operator}$

e.g:
 $\text{int } x = 10;$
 $\text{int } *p = \&x;$
 ↑
 Refer

2) Dereference operator:

$\text{int } i \text{ is also known as value operator}$

e.g:-
 $\text{int } x = 10;$
 $\text{int } *p = \&x;$
 Deref
 printf("%d", *p); // 10

Pointer arithmetic

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n1,n2,ans;
    int *p1,*p2;
    printf("Enter two numbers\n");
    scanf("%d %d", &n1, &n2);
    p1=&n1;
    p2=&n2;
    ans=*p1 + *p2;
    printf("Addition=%d", ans);
    getch();
}
```

int n1,n2,ans;
int *p1,*p2;

printf("Enter two numbers\n");

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

p1=&n1;

p2=&n2;

ans=*p1 + *p2;

printf("Addition=%d", ans);

getch();

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

Pointers :- Pointer is variable which is used to store the address of another variable.

Syntax

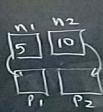
```
data-type *pointer-name;  
e.g:- int *p;  
      char *c;  
      float *ptr;
```



1) Reference operator :-

int x; // 10
printf("%d", x); // 10
printf("%d", *p); // 10
e.g:-
int x = 10;
int *p = &x;

$\text{int } x \text{ is also known as address operator}$



2) Dereference operator :-

int x = 10;
int *p = &x;
printf("%d", *p); // 10
e.g:-
int x is also known as value operator

Pointer arithmetic

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
    int n1,n2,ans;  
    int *p1,*p2;  
    printf("Enter two numbers\n");  
    scanf("%d %d", &n1, &n2);  
    p1=&n1;  
    p2=&n2;  
    ans=*p1 + *p2;  
    printf("Addition=%d", ans);  
    getch();
```

Write QP of following program

```
void main()  
{  
    int x=0,y,*ip;  
    ip=&x;  
    y=(*ip)++;  
    printf("%d\n", y);  
    printf("%d\n", *ip);  
    y=++x (*ip),  
    printf("%d\n", y);  
    printf("%d\n", *ip);  
    getch();  
}
```

Write off the following program

void main()

{ int x=20, y, *ip;

ip = &x;

y = (*ip)++,
20

printf("%d\n", y); // 20

12), printf("%d\n", *ip); // 21

y = ++x (*ip);

printf("%d\n", y); // 22

printf("%d\n", *ip); // 22

} return();

22

22

1010

x
1010

y

ip

// 20

// 21

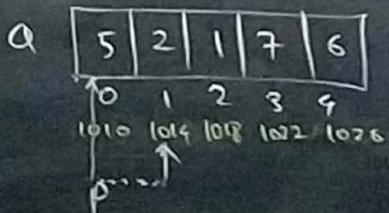
22

22

Pointer to array

Pointer stores the address of array.

WAP to display sum of array elements using pointer



Pointer

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[10], n, i, sum=0, *p;
    printf("Enter no of elements\n");
    scanf("%d", &n);
    printf("Enter the values\n");
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    p=a;
    for(i=0, i<n, i++)
        sum = sum + *(p+i);
    printf("Sum=%d", sum);
}
```

Dynamic memory allocation

Memory is allocated dynamically during execution of program

These is available in "stdlib.h"

- 3) free(): It is used to free or deallocate memory allocated dynamically

Syntax:

free(pointer);

ptr = (typecast)malloc(Size in bytes);

e.g:-

int * ptr;

ptr = (int*) malloc(10 * sizeof(int))

- 2) calloc(): It is used to allocate memory dynamically with initial value as "zero"

Syntax:

ptr = (typecast)calloc(Size in bytes);

```
Polish
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, *p, i;
    print("Enter no of elements");
    scanf("%d", &n);
    p=(int*)malloc(n * sizeof(int));
    for(i=0; i<n; i++)
    {
        scanf("%d", (p+i));
    }
    for(i=0; i<n; i++)
    {
        print("%d", *(p+i));
    }
}
```

FILES :-

FILE is group of bytes used to store information on secondary storage device

File pointer :

FILE *ptr-name;

Operations

1. Open file

2. Read

3. Write

4. Close

1) Open file

fopen() is function used to open the file.

Syntax

ptrname=fopen("NameOfFile", "mode");

Different modes of opening File

1) "r" : It is read only mode

If file is present then it open for reading

If file is not present it returns NULL ..

Ex :-

ptr = fopen("test.txt", "r");

"w" : It is write only mode

If file is present then its contents are destroyed

If file is not present then it creates file.

Ex :-
ptr=fopen("test.txt", "w")

"a" : It is append only mode

If file is present then its content are maintained & writing will start at end of existing

If file is not present then it's create file with same name

Ex :-

ptr=fopen("test.txt", "a")

2) "r+" : additionally it can write

3) "wt" : additionally it can write

4) "at" : additionally it can read.

Read / write data into file

1) `fgetc()`: It is used to read data from file character by character.

Syntax

`fget(Fpointer);`

2) `fgets()`: It is used to read data from file.
It can read collection of character.

Syntax

`fgets(string, int n, File pointer);`

3) `fscanf()`: It can read any type of data from file.

Syntax

`fscanf(Filepointer, "format", "Address of Variable")`

Eg:-
`fscanf(fp, "%d %f", &x, &y);`

Writing

1) `fputc()`: It writes data into file character by character.

Syntax

`fputc(char, Fpointer)`

2) `fputs()`: It writes data into file.
It can write collection of character.

Syntax

`fputs(string, File pointer);`

3) `fprintf()`: It can write any type of data to file.

Syntax

`fprintf(fp, "format", Variable);`

Eg:-

`fprintf(fp, "%d %f", x, y);`

Write a program to read & write data into file.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    FILE *Fptr;
    char ch;
    Fptr = fopen("test.txt", "w");
    printf("Enter data to write into file once completed
           enter full stop( . )\n");
    while(ch != '.')
        scanf("%c", &ch);
        fputc(ch, Fptr);
    fclose(Fptr);
    Fptr = fopen("test.txt", "r");
}
```

```
while(!feof(Fptr))
{
    ch = fgetc(Fptr);
    printf("%c", ch);
}
fclose(Fptr);
getch();
}
```

Algorithm :-
Step by step instruction written in
human understandable language

Properties :-

- Non Ambiguity
- Range of Inputs
- Multiplicity
- Definiteness

Addition of two numbers

Step 1 : Start

Step 2 : INPUT n_1, n_2 ,

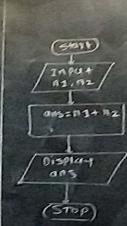
Step 3 : $ans = n_1 + n_2$;

Step 4 : DISPLAY ans

Step 5 : STOP

Flowchart :- graphical representation
of algorithm

sum/end
Input/Output
Assignment/Process
Flow Lines
Condition
(if, for, while)



While computation, break fractions
to compute two numbers

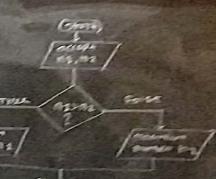
Step 1 : Start

Step 2 : INPUT n_1, n_2 ,

Step 3 : $ans = n_1 + n_2$;

Step 4 : DISPLAY ans

Step 5 : STOP



Flowchart :- Graphical representation of algorithm

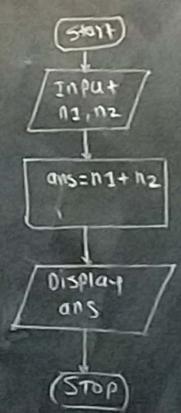
Start/End

Input/Output

assignment/process

Flow Lines

condition
(if, for, while)



Write algorithm & draw flowchart to compare two numbers

Step 1: Start

Step 2: Accept n_1 & n_2

Step 3: If $n_1 > n_2$ then

display n_1 is max

else

display n_2 is max

endif

Step 4: Stop,

