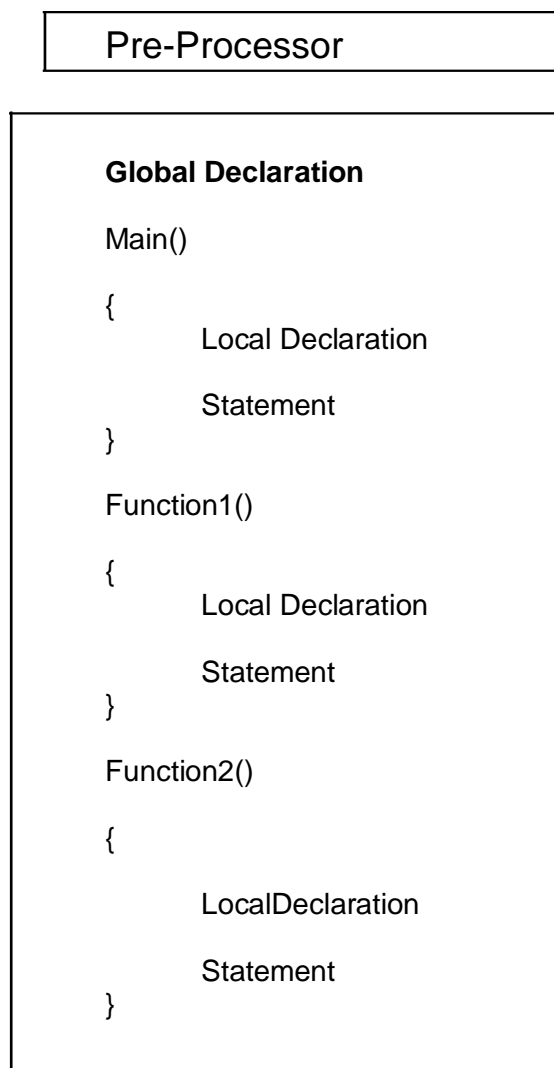


INTRODUCTION TO 'C'

The programming language developed in the early 1970s by Dennis Ritchie at Bell Lab to be used by the Unix Operating System. It is a high level language enables the programmer to concentrate on the problem about the machine code on which the program would be run. 'C' language is primarily used for system programming. It is a portable language.

STRUCTURE OF A 'C' PROGRAM



PRE-PROCESSOR DIRECTIVES

`#include<stdio.h>` : -This is a pre-processor command that comes as the first statement in our code. All pre-processor command start with '#' symbol. The ' #' include statement tells the compiler to include the standard I/O library or header file(`stdio.h`) in the program.

`Main()` : - It is the main function where all 'C' program should be kept inside.

SYNTAX FOR WRITING A 'C' PROGRAM

```
#include<stdio.h>

void main()

{
    _____ statements;
    _____ statements;
}
```

DATA TYPES IN 'C'

- Integer
- Float
- Double
- Character
- String

SPECIFIER USED IN 'C'

- `int-%d`
- `float-%f`
- `double-%ld`
- `character-%c`
- `string-%s`

INPUT /OUTPUT STATEMENT IN 'C'

`printf("message");` - It is the output command.

scanf("specifier", &variable name); - It is the input command.

COMPILE AND RUN THE 'C' PROGRAM

Compile–Alt+F9

Run –Ctrl+F9

Result–ALT+F5

CONTROL STATEMENT IN 'C'

- Sequential statement
- Selection/Branching statement
- Iteration /Looping statement

Sequential statement

Here all statements are execute line by line i.e. sequentially.

Simple Program

Q1.W.A.P to print the message as “This is my 1stprogram”.

```
#include<stdio.h>
#include<conio.h>
void main()
{
printf(“This is my 1stprogram”);
}
```

Output =

This my 1st program.

Q2.W.A.P to accept sum of 2numbers.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    Int a, b, sum;
    clrscr();
    printf(“enter two nos”);
    scanf(“%d%d”,&a,&b);
    sum=a+b;
    printf(“The sum of the two number is %d”, sum);
}
```

Output =

```
Enter two numbers    23   24
Sum of numbers is    47
```

Q3.W.A.P to accept two numbers swap their values without using 3rd variable.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a,b;
    clrscr();
    printf(“enter two number”);
    scanf(“%d %d”, &a, &b);
```

```

    a=a+b;
    b=a-b;
    a=a-b;
    printf("after swapping values are %d\n%d",a,b);
    getch();
}

```

Output =

```

Enter two numbers      34    54
After swapping         54    34

```

Q4.W.A.P to find the simple interest.

```

#include<stdio.h>
#include<conio.h>
void main()
{
    float p,r,t, si;
    clrscr();
    printf("enter principle, rate, time");
    scanf("%f%f%f",&p,&r,&t);
    si=(p*r*t)/100;
    printf("simple interest is%f",si);
    getch();
}

```

Output =

```

Enter principle, rate, time      1000      0.03      2
The simple interest is 6.000

```

Selection / Branching statement

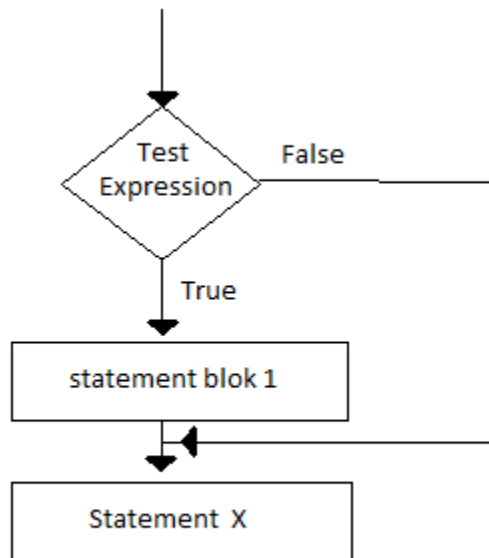
Here the statements are execute depends upon their condition.

- IF statement
- Switch case statement

IF STATEMENT – It is the simplest form of decision control statements that is frequently used in decision making. The general form of a simple if statement is

SYNTAX OF IF – ELSE STATEMENT

```
If(test expression)
{
    statement1;
    .....
    statement n;
}
statement x;
```

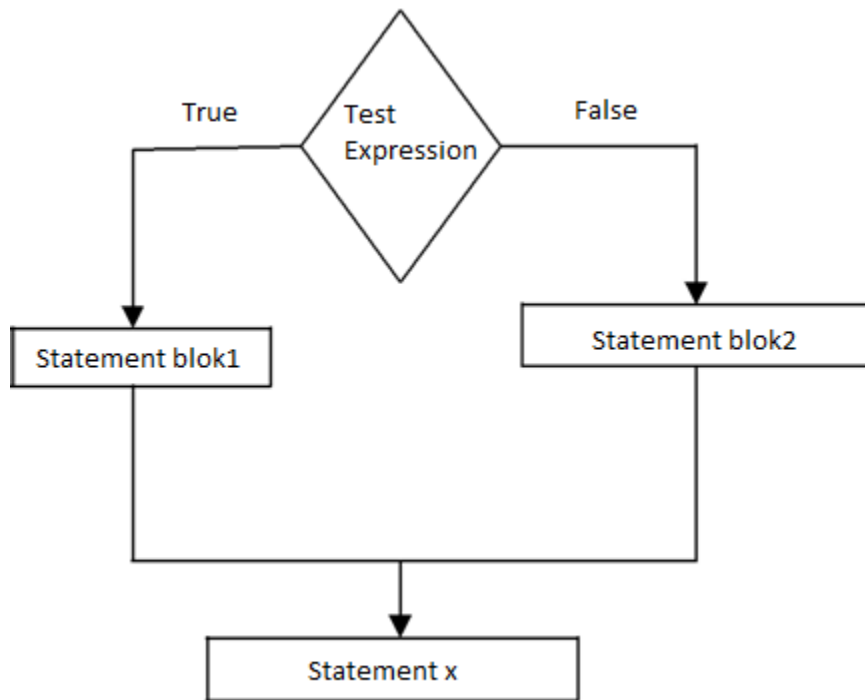


IF ELSE STATEMENT– It plays a vital role in conditional branching. It's usage is very simple. The test expression is evaluated, if the result is true, then statement(s) followed by the expression is executed else, if the expression is false, the statement is skipped by the compiler.

SYNTAX OF IF–ELSE STATEMENT

```
If(test expression)
{
    Statement Block1;
}
else
{
```

```
Statement Block2;  
}  
  
Statement X;
```



SWITCH CASE

A switch case statement is a multi-way decision statement that is a simplified version of an if-else block that evaluates only one variable.

SYNTAX OF SWITCH STATEMENT

```
switch(variable)  
{  
    Case value1:  
        Statement Block 1;  
        Break;  
    Case value2:
```

Statement Block 2;

Break;

.....

Case valueN:

Statement Block N;

Break;

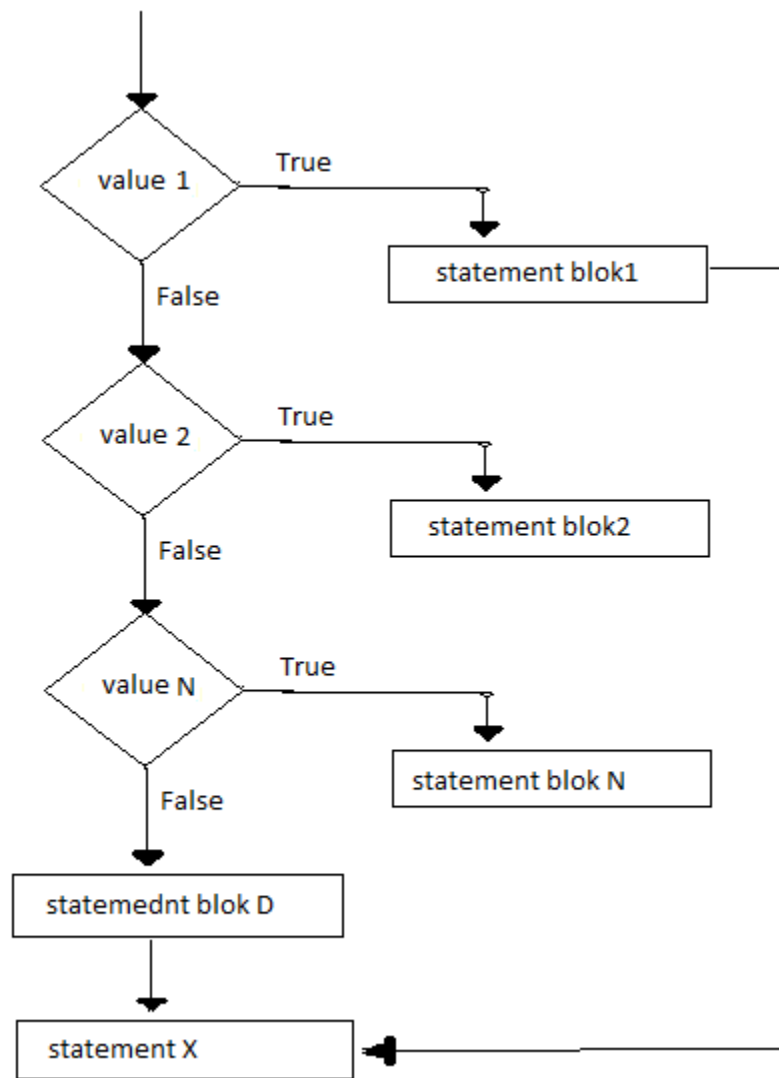
Default:

Statement Block D;

Break;

}

Statement X;



Selection / Branching Program

Q1.W.A.P a program to accept two numbers and find the smallest number.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    Int a,b;
    clrscr();
    printf("enter two numbers");
    scanf("%d%d", &a,&b);
    if(a<b)
    {
        printf("smallest is%d",a);
    }
    else
    {
        printf("smallest is%d",b);
    }
}
```

Output =

Enter two number 57 56

Q2.W.A.P to accept 3 numbers and find the largest one.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    Int a,b,c;
    clrscr();
    printf("enter3nos");
    scanf("%d%d%d",&a,&b,&c);
    if(a>=b && a>=c)
    {
        printf("largest is%d",a);
    }
    else
    if(b>=a && b>=c)
    {
        printf("largest is%d",b);
    }
}
```

```

        else
        if(c>=a && c>=b)
        {

                printf("largest is%d",c);

        }
}

```

Output =

```

Enter3nos 34 90 70
Largest is 90

```

Q3. W.A.P to check whether it is positive, negative or zero.

```

#include<stdio.h>
#include<conio.h>
void main()
{
        Int a;
        clrscr();
        printf("enter a no");
        scanf("%d",&a);
        if(a>0)
        {
                printf("no.ispositive");
        }
        else
        if(a<0)
        {
                printf("no.isnegative");
        }
        else
        if(a==0)
        {
                printf("no.iszero");
        }
        getch();
}

```

Output =

```

Enter a no 6
no.is positive

```

Q4. Write a program to accept a number to check whether it is even or odd.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    Int a;
    clrscr();
    printf("enter a no.");
    scanf("%d",&a);
    if(a%2==0)
    {
        printf("no.is even");
    }
    else
    if(a%2!=0)
    {
        printf("no. is odd");
    }
    getch();
}
```

Output =

Enter a no. 35
no.is odd

Q5.W.A.P to accept the day code, print the day name.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char day;
    clrscr();
    printf("enter day code M,T,W,H,F,S,U");
    scanf("%c",&day);
    switch(day)
    {
        case 'M':
            printf("Monday");
            break;
        case 'T':
```

```
        printf("Tuesday");
        break;
    case 'W':
        printf("Wednesday");
        break;
    case 'H':
        printf("Thursday");
        break;
    case 'F':
        printf("Friday");
        break;
    case 'S':
        printf("Saturday");
        break;
    default :
        printf("wrong day code");
    }
    getch();
}
```

Output =

Enter day a code M,T,W,H,F,S,U M
Monday.

ITERATIVE STATEMENTS

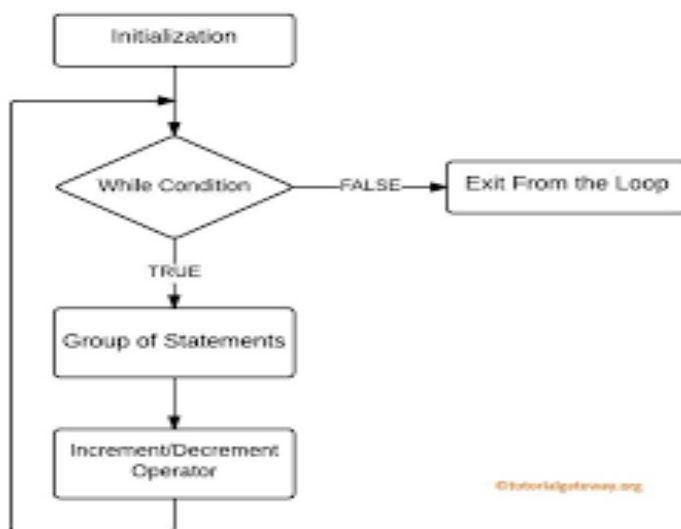
Iterative Statements are used to repeat the execution of a list of statements, depending on the value of an integer expression. 'C' language supports 3 types of iterative statements also known as looping statements. They are:-

- While loop
- do-while loop
- for loop

WHILELOOP:-In the while loop, the condition is tested before any of the statement in the statement block is executed. If the condition is true, only then the statements will be executed otherwise if the condition is false, the control will jump to statement Y, which is the immediate statement outside the while loop block.

SYNTAX OF WHILELOOP

```
Statement X;  
while(condition)  
{  
    Statement block;  
}  
Statement Y;
```



DO WHILE LOOP:-

The do while loop is similar to the while loop .The only difference is that in a do–while loop, the test condition is tested at the end of the loop.

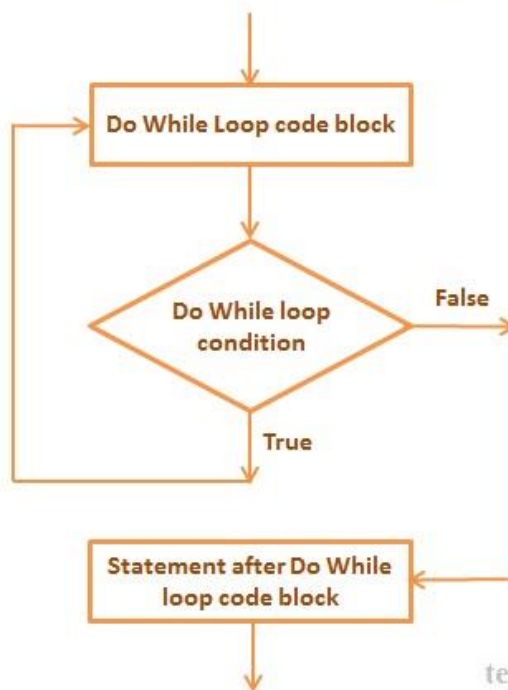
SYNTAX OF DO–WHILELOOP

Statement X;

```
do
{
Statement block;
}
while(condition);
```

statement Y;

Do While Loop Flow Diagram



FOR LOOP:-The For Loop provide particular condition is true .For Loop is usually known as a determinate or definite loop because the programmer knows exactly how many times the loop will repeat. The number of times the loop has to be executed can be determined mathematically by checking the logic of the loop.

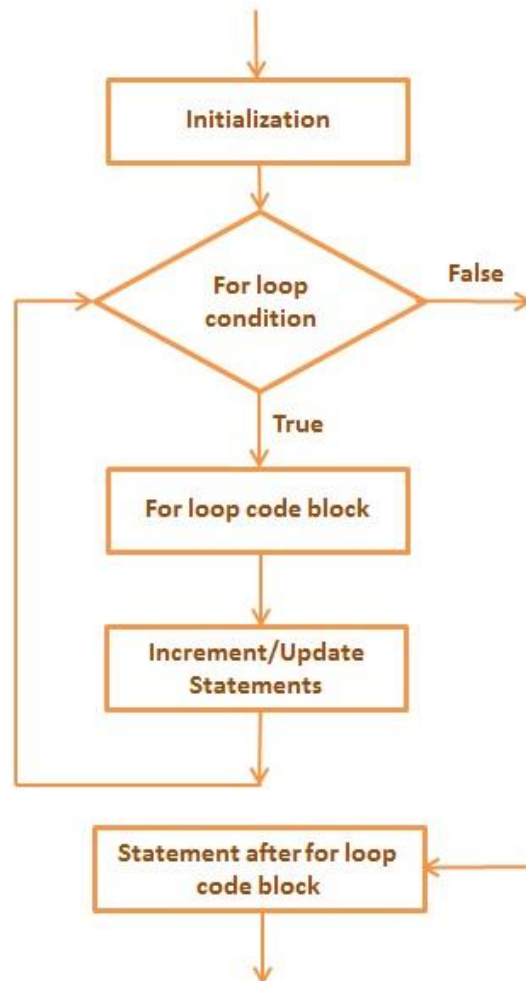
SYNTAX OF FOR LOOP

For(initialization ; condition ;increment/decrement/update)

```
{  
    Statement block;  
}
```

Statement Y;

For Loop Flow Diagram



techcrashcourse.com

LOOPING PROGRAMS

Q1) WAP TO PRINT ALL NATURAL NO FROM 1-10 BY USING FOR LOOP

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i;
    clrscr();

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

OUTPUT =

1 2 3 4 5 6 7 8 9 10

Q2) To display our college name twenty times on screen by using while.

```
#include<stdio.h>
#include<conio.h>
voidmain()
{
    Int i=1;
    clrscr();
    while (i <=20)
    {
        printf("\n S.K.D.A.V ");
        i = i+1;
    }
}
```

OUTPUT =

S.K.D.A.V	S.K.D.A.V	S.K.D.A.V	S.K.D.A.V	S.K.D.A.V
S.K.D.A.V	S.K.D.A.V	S.K.D.A.V	S.K.D.A.V	S.K.D.A.V
S.K.D.A.V	S.K.D.A.V	S.K.D.A.V	S.K.D.A.V	S.K.D.A.V
S.K.D.A.V	S.K.D.A.V	S.K.D.A.V	S.K.D.A.V	S.K.D.A.V

Q3) WAP to print all even no (1-100) by using do...while

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int i=1;
    clrscr();
    do
    {
        if(i%2==0)
        {
            printf("\n%d",i);
        }
        i++;
    }while(i<=100);
    getch();
}

```

OUTPUT =

2	4	6	8	10	12	14	16	18		
20	22	24	26	28	30	32	34	36	38	
40	42	44	46	48	50	52	54	56	58	
60	62	64	66	68	70	72	74	76	78	
80	82	84	86	88	90	92	94	96	98	
100										

Q4) to perform addition of 1-100 number

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int i,sum=0;
    clrscr();
    for(i=1; i<=100; i++)
    {
        sum = sum +i;
    }
    printf(" sum of all no %d", sum);
    getch();
}

```

OUTPUT =

Sum of all no 5050

ARRAY

An array is a collection of similar data elements. These data elements have same data types. The element of the array are store din consecutive memory ,location & are referenced by an index (also known as the subscript).Subscripts indicates at ordinal no. of the element counted from the beginning of the array.

DECLARATION OF ARRAY

An array must be declared before being used. Declaring an array means specifying things:-

DATATYPE: What type of data it can be stored.

For eg– int , char , float , double.

NAME : To identify the array.

SIZE :The maximum no .of value that the array can hold.

SYNTAX FOR DEFINE AN ARRAY

Type name [size];

Eg–int marks [10];

Q1) W.A.P to store 10 student mark.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,mark[10];
    clrscr();
    printf("enter 10 elements");
    for(i=0;i<10;i++)
    {
        scanf("%d",&mark[i]);
    }
    printf("elements are:\n");
    for(i=0;i<10;i++)
    {
        printf("%d\t",mark[i]);
    }
    getch();
}
```

OUTPUT =

```
enter 10 elements 1    2    3    4    5    6    7    8    9    10
elements are:
                1    2    3    4    5    6    7    8    9    10
```

Q2) W.A.P to store 10 student mark find its sum and avg.

```

#include<stdio.h>
#include<conio.h>
void main()
{
    float i,mark[10],sum,avg;
    clrscr();
    printf("enter 10 elements");

    for(i=0;i<10;i++)
    {
        scanf("%f",&mark[i]);
        sum += mark[i];
    }
    avg = sum/10;

    printf("sum is %f",sum);
    printf("avg is %f",avg);
    getch();
}

```

OUTPUT =

```

enter 10 elements
          10   20   10   5    15   10   5    10   20   30
sum is 135.000000avg is 13.500000

```

Q3) W.A.P to Find smallest / largest number from array elements.

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int i,mark[10],lar,small;
    clrscr();
    printf("enter 10 students mark");
    for(i=0;i<10;i++)
    {
        scanf("%d",&mark[i]);
    }
    lar=mark[0];
    small=mark[0];
    for(i=0;i<10;i++)
    {
        if(mark[i] > lar)
        {
            lar=mark[i];

```

```

    }
}
for(i=0;i<10;i++)
{
    if(mark[i] < small)
    {
        small=mark[i];
    }
}
printf("large no. is %d \n",lar);
printf("small no is  %d",small);
getch();
}

```

OUTPUT =

```

enter 10 students mark
    98  90  87  78  65  67  93  58  545  34
large no. is 545
small no is 34

```

Q4) W.A.P to sort the elements of an array in ascending order.

```

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

void main()
{
    int i,j,temp,mark[10];
    clrscr();
    printf("enter 10 students mark");

    for(i=0;i<10;i++)
    {
        scanf("%d",&mark[i]);
    }

    for(i=0;i<10;i++)
    {
        for (j=0; j<10; j++)
        {
            if(mark[j] > mark[j+1])
            {
                temp = mark[j];

```

```

        mark[j] = mark[j+1];
        mark[j+1] = temp;
    }
}
printf("ascending order are ");
for(i=0; i<10; i++)
{
    printf("%d\n", mark[i]);
}
getch();
}

```

OUTPUT =

enter 10 students mark

4 56 78 2 90 567 43 87 66

ascending order are

2 3 4 4 56 66 78 87 90 567

Q5) W.A.P to sort the elements of an array in descending order.

```

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

void main()
{
    int i,j,temp,mark[10];
    clrscr();
    printf("enter 10 students mark");

    for(i=0;i<10;i++)
    {
        scanf("%d",&mark[i]);
    }

    for(i=0;i<10;i++)
    {
        for (j=0; j<10; j++)
        {
            if(mark[j] < mark[j+1])
            {
                temp = mark[j];
                mark[j] = mark[j+1];
            }
        }
    }
}

```

```

        mark[j+1] = temp;
    }
}
printf("descending order are ");
for(i=0; i<10; i++)
{
    printf("%d\n", mark[i]);
}
getch();
}

```

OUTPUT =

enter 10 students mark

4 56 78 2 90 567 4 3 87 66

descending order are

567 90 87 78 66 56 4 4 3 2

Q6) W.A.P to enter elements for 3x3 matrix and display them.

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j,mark[3][3];
    clrscr();
    printf("enter 9 elements ");
    for(i=0;i<3;i++)
    {
        for (j=0; j<3; j++)
        {
            scanf("%d",&mark[i][j]);
        }
    }
    printf("\n the matrix elements are \n");
    for(i=0;i<3;i++)
    {
        for (j=0; j<3; j++)
        {
            printf("%d \t",mark[i][j]);
        }
        printf("\n");
    }
}

```

OUTPUT =

Enter 9 elements

11	22	33
44	55	66
77	88	99

the matrix elements are

11	22	33
44	55	66
77	88	99

Q7) WAP to find addition of 2 dimensional matrix
//first array row + second array row

```
#include <stdio.h>
void main()
{
    int i, j, first[2][2], second[2][2], sum[2][2];
    clrscr();
    printf("Enter the elements of first matrix\n");
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 2; j++)
        {
            scanf("%d", &first[i][j]);
        }
    }
    printf("Enter the elements of second matrix\n");
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 2; j++)
        {
            scanf("%d", &second[i][j]);
        }
    }

    printf("\n\t\t");

    printf("\n OUTPUT OF first matrix\n");
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 2; j++)
        {
            printf("\t[%d][%d]  %d", i, j, first[i][j]);
        }
    }
```



```

        printf("\n");
    }
    printf("\n OUTPUT OF second matrix\n");
    for (i = 0; i < 2; i++)
    {
        for (j = 0 ; j < 2; j++)
        {
            printf("\t[%d][%d]  %d", i,j,second[i][j]);
        }
        printf("\n");
    }

    printf("Sum of entered matrices:-\n");

    for (i = 0; i < 2; i++)
    {
        for (j = 0 ; j < 2; j++)
        {
            sum[i][j] = first[i][j] + second[i][j];
            printf("%d\t", sum[i][j]);
        }
        printf("\n");
    }
}

```

OUTPUT =

Enter the elements of first matrix

1
2
3
4
Enter the elements of second matrix
5
6
7
8

OUTPUT OF first matrix

[0][0]	1	[0][1]	2
[1][0]	3	[1][1]	4

OUTPUT OF second matrix

[0][0]	5	[0][1]	6
--------	---	--------	---

[1][0] 7 [1][1] 8
Sum of entered matrices:-
6 8
10 12

Q8) WAP to find Substraction of 2 dimensional matrix
//first array row - second array row

```
#include <stdio.h>

void main()
{
    int i, j, first[2][2], second[2][2], diff[2][2];
    clrscr();
    printf("Enter the elements of first matrix\n");
    for (i = 0; i < 2; i++)
    {
        for (j=0; j<2; j++)
        {
            scanf("%d", &first[i][j]);
        }
    }
    printf("Enter the elements of second matrix\n");
    for (i=0; i<2; i++)
    {
        for (j=0 ; j<2; j++)
        {
            scanf("%d", &second[i][j]);
        }
    }

    printf("\n\t\t");

    printf("\n OUTPUT OF first matrix\n");
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 2; j++)
        {
            printf("\t[%d][%d]  %d", i, j, first[i][j]);
        }
        printf("\n");
    }
    printf("\n OUTPUT OF second matrix\n");
```

```

for (i = 0; i < 2; i++)
{
    for (j = 0 ; j < 2; j++)
    {
        printf("\t[%d][%d]  %d", i,j,second[i][j]);
    }
    printf("\n");
}

printf("Sum of entered matrices:-\n");

for (i = 0; i < 2; i++)
{
    for (j = 0 ; j < 2; j++)
    {
        diff[i][j] = first[i][j] - second[i][j];
        printf("%d\t", diff[i][j]);
    }
    printf("\n");
}
}

```

Output=

Enter the elements of first matrix

10

9

14

60

Enter the elements of second matrix

5

2

3

6

OUTPUT OF first matrix

[0][0]	10	[0][1]	9
[1][0]	14	[1][1]	60

OUTPUT OF second matrix

[0][0]	5	[0][1]	2
[1][0]	3	[1][1]	6

Diff of entered matrices:-

5 7

11 54

Q9) W.A.P to calculate multiplication of 2 dimensional matrix.

```

#include<stdio.h>
#include<conio.h>
void main ()
{
int a[3][3],b[3][3],c[3][3],i,j,k;

printf("enter the value of a");
for(i=0;i<3;i++)
{
for(j=0;j<3; j++)
{
scanf("%d",&a[i][j]);
}
}
printf("enter the value of b");
for (i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
scanf("%d",&b[i][j]);
}
}

for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
c[i][j]=0;
for(k=0;k<3;k++)
{
c[i][j] = c[i][j]+a[i][k]*b[k][j];
}
}
}
printf("the result is");
printf("\n");
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
printf("%d", c[i][j]);
}
}

```

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

Output

Enter the value of a

1 3 5

8 2 4

6 9 7

Enter the value of b

2 2 3

4 1 2

6 2 1

The result is

44 15 14

48 26 32

90 35 43

STRINGS

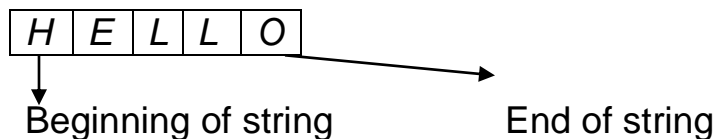
In 'C' language , a string is nothing but a null terminated character array. This means that after the last character, a null character ('\0') is stored to signify the end of the character array.

Ex. If we write

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

```
char str[5] = "HELLO" ;
```

```
char [ ] = "HELLO" ;
```



str[0]	1000	H
str[1]	1000	E
str[2]	1000	L
str[3]	1000	L
str[4]	1000	O
str[5]	1000	'\0'

The five commonly used string handling function are
strcat(), strcmp(), strcpy(), strlen(), strrev()

To demonstrate output of standard library function

Q1) WAP to print a string print it

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char sen[10];
clrscr();
printf("enter the string");
scanf("%s",sen);
printf("the string is %s",sen);
getch();
}
```

OUTPUT =

enter the string rourkela
the string is rourkela

Q2) WAP TO PRINT THE STRING IN LOWER CASE (STRLWR)

```
#include<stdio.h>
#include<conio.h>
main()
{
char str[10];
clrscr();
printf("Enter a string");
gets(str);
printf("string in lowercase is %s",strlwr(str));
}
```

OUTPUT =

Enter a stringRKL
string in lowercase is rkl

Q3) WAP TO PRINT THE STRING IN UPPERCASE (STRUPR)

```
#include<stdio.h>
#include<conio.h>
main()
{
char str[10];
clrscr();
printf("Enter a string");
gets(str);
```

```
printf("string in lowercase is %s",strupr(str));
```

```
}
```

OUTPUT =

Enter a string rkl

string in lowercase is RKL

Q4) WAP TO PRINT THE STRING IN REVERSE CASE (STRREV)

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

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

```
main()
```

```
{
```

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

```
clrscr();
```

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

```
gets(str);
```

```
printf("string in lowercase is %s",strrev(str));
```

```
}
```

OUTPUT =

Enter a string rourkela

string in lowercase is alekruor

Q5) WAP TO APPEND ONE STRING TO ANOTHER STRING (STRCAT)

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

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

```
void main()
```

```
{
```

```
char str1[10], str2[10];
```

```
clrscr();
```

```
printf("Enter a 1st string    : ");
```

```
gets(str1);
```

```
printf("Enter a 2nd string    : ");
```

```
gets(str2);
```

```
printf("Append one string to another is %s",strcat(str1,str2));
```

```
}
```

OUTPUT =

Enter a 1st string : rkl

Enter a 2nd string : ctc

Append one string to another is rklctc

Q6) WAP TO APPEND ONE STRING TO ANOTHER STRING (STRCPY)

```
#include<stdio.h>
#include<conio.h>
void main()
{
char str1[10], str2[10];
clrscr();
printf("Enter a 1st string  : ");
gets(str1);
printf("Enter a 2nd string  : ");
gets(str2);
printf("Copy one string to another is %s",strcpy(str1,str2));
}
```

OUTPUT =

```
Enter a 1st string  : rkl
Enter a 2nd string  : ctc
Copy one string to another is ctc
```

FUNCTION

A function is a sub-program which can act on data and produce a result. In C the function should be define before it's usage in the program.

Syntax of function

Declaration section

<return type> function nm(paramenter list);

Definition section

<return type> function nm(parameter list)

{

Body of the function

}

Function call

Function nm(parameter);

Every 'C' programme has atleast on main function that is main(). When the program start, the function main() is automatically called and this can call other function, but it can't be called by any other function.

Functions are of two types

- a) Built in Functions
- b) User defined Functions

Built in Functions are called library functions and they are the part of compiler package. User defined functions are used according to the requirement of the program and created by the program himself.

Functions are called by two method

- a) Call by value
- b) Call by reference

Call by value : In Function values of the variables are passed by the calling function to the called function. The Programs that we have written so far call the function using call by value method of passing parameters.

Call by reference : In which address of the variables are passed by the calling function to the called functions.

Recursion Function

A recursive function is a function that calls itself during its execution. This enables the function to repeat itself several times, outputting the result and the end of each iteration.

Q1) TO CALCULATE AREA OF A CIRCLE BY USING FUNCTION

```
void circle(int rad);
#include <stdio.h>
void main()
{
    float r;
    clrscr();
    printf("Enter the radius of circle : ");
    scanf("%f", &r);
    circle(r);
}
void circle(float rad)
{
    float area;
    area = 3.14 * rad * rad;
    printf("Area of circle : %f", area);
}
```

OUTPUT =

```
Enter the radius of circle : 3
Area of circle : 28.260000
```

Q2) TO CALCULATE FACTORIAL OF A GIVEN NO USING RECURSION.

```

#include <stdio.h>
long int fact(int n);
void main()
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Factorial of %d = %ld", n, fact(n));
}
long int fact(int n)
{
    if (n >= 1)
        return n * fact(n-1);
    else
        return 1;
}

```

OUTPUT =

Enter a positive integer : 4
 Factorial of 4 = 24

Q3) to demonstrate call by reference , call by value.

```

void swap(int x, int y);
#include <stdio.h>

void main()
{
    int a = 10, b = 20;
    clrscr();
    printf("\n BEFORE SWAP VALUES ARE a=%d b=%d", a, b);
    swap(a,b);
    printf("\n AFTER SWAP IN MAIN FUN VALUES ARE a=%d b=%d ", a, b);
}

void swap(int x, int y)
{
    int temp;

    temp = x;
    x = y;
    y = temp;

    printf("\n INSIDE SWAP FUNCTION VALUES ARE x=%d y=%d ", x, y);
}

```

OUTPUT =

BEFORE SWAP VALUES ARE a=10 b=20
INSIDE SWAP FUNCTION VALUES ARE x=20 y=10
AFTER SWAP IN MAIN FUN VALUES ARE a=10 b=20

Q4) C program to illustrate Call by Reference

```
void swap(int *, int *);  
#include <stdio.h>  
void main()  
{  
    int a = 10, b = 20;  
    clrscr();  
    printf("\n BEFORE SWAP VALUES ARE a=%d  b=%d",a , b);  
    swap(&a, &b);  
    printf("\n AFTER SWAP IN MAIN FUN VALUES ARE a=%d b=%d", a, b);  
}  
  
void swap(int *x, int *y)  
{  
    int temp;  
    temp = *x;  
    *x = *y;  
    *y = temp;  
    printf("\n AFTER SWAP INSIDE SWAP FUN VALUES ARE  x=%d y=%d",*x,  
*y);  
}
```

OUTPUT =

BEFORE SWAP VALUES ARE a=10 b=20
INSIDE SWAP FUNCTION VALUES ARE x=20 y=10
AFTER SWAP IN MAIN FUN VALUES ARE a=20 b=10

STRUCTURE

Structure is basically a user defined data type that can store related information (even if different data types) together.

A structure is therefore a collection of variables under a single name. The variables within a structure are of different data types and each has a name that is used to select it from the structure.

Syntax

```
struct <structure tag name>
{
    datatype <variable nm>;
    datatype <variable nm>;
    -----
} <structure variable nm>;
```

Example:

Struct student

```
{
int rno ;
char nm[10];
float mk;
} std;
```

To maintain & manipulate student data using structure.

Q1) WAP to store students Rno, Nm and Mark in a structure& print it.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    struct student
    {
        int rollno;
        char nm[10];
        int mark;
    } std;

    int i;
    clrscr();
    printf("enter the roll no");
    scanf("%d",&std.rollno);
    printf("enter the name");
    scanf("%s",std.nm);
    printf("enter the mark");
    scanf("%d",&std.mark);

    printf("\n roll no %d",std.rollno );
    printf("\t name %s",std.nm);
    printf("\tmark is %d",std.mark);
```

```
}  
}
```

OUTPUT =

```
enter the rollno  101  
enter the name   ajay  
enter the mark   78  
roll no 101      name ajay mark is    78
```

Q2) WAP to store a single students rno,nm and 3 sub mark in a structure.

```
#include<stdio.h>  
#include<conio.h>  
void main()  
{  
    struct student  
    {  
        int rollno;  
        char nm[10];  
        int mark[3];  
    } std;  
    int i;  
    clrscr();  
    printf("enter the roll no");  
    scanf("%d",&std.rollno);  
    printf("enter the name");  
    scanf("%s",std.nm);  
    printf("enter the mark");  
    for(i=0;i<3;i++)  
    {  
        scanf("%d",&std.mark[i]);  
    }  
    printf("\nroll no %d",std.rollno );  
    printf("\nname %s",std.nm);  
    for(i=0; i<3; i++)  
    {  
        printf("mark is %d",std.mark[i]);  
    }  
}
```

OUTPUT =

```
enter the rollno  101  
enter the name   ajay  
enter the mark   78   93   99  
  
roll no 101      name ajay mark is    95   93   99
```

Q3) WAP to store five students RNO, NAME and MARK in a structure.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    struct student
    {
        int rollno;
        char nm[10];
        int mark;
    }std[5];

    int i;
    clrscr();

    for(i=0;i<5;i++)
    {
        printf("enter the roll no");
        scanf("%d",&std[i].rollno);
        printf("enter the name");
        scanf("%s",std[i].nm);
        printf("enter the mark");
        scanf("%d",&std[i].mark);
    }
    for(i=0;i<5;i++)
    {
        printf("roll no %d",std[i].rollno );
        printf("name %s",std[i].nm);
        printf("mark is %d",std[i].mark);
    }
    printf("\n");
}
```

OUTPUT =

```
enter the rollno   101
enter the name     ajay
enter the mark     78
enter the roll no  102
enter the name     biju
enter the mark     90
enter the roll no  103
```

```
enter the name ram
enter the mark 99
enter the roll no 104
enter the name gopal
enter the mark 98
enter the roll no 105
enter the name jay
enter the mark 97
```

```
roll no 101 name ajay mark is 78
roll no 102 name biju mark is 90
roll no 103 name ram mark is 99
roll no 104 name gopal mark is 98
roll no 105 name jay mark is 97
```

POINTER

A pointer provides access to a variable by using the address of that variable. A pointer variable is therefore a variable that stores the address of another variable.

```
datatype *ptr name ;
int *ptrnm;
```

The '*' informs the compiler that ptr is a pointer variable and the int specifies that it will store the address of an integer variable. An integer pointer variable therefore, points to an integer variable.

Q1) To perform 4 arithmetic functions on pointers.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int *ptr1,*ptr2;
int a=15,b=5;
clrscr();
ptr1 = &a;
ptr2 = &b;
printf("\n addition of 2 nos is %d",(*ptr1 + *ptr2));
```

```
printf("\n subtraction  of 2 nos is %d",(*ptr1 - *ptr2));  
printf("\n multiplication of 2 nos is %d",(*ptr1 * *ptr2));  
printf("\n division  of 2 nos is %d",(*ptr1 / *ptr2));  
}
```

OUTPUT =

addition of 2 nos is 20
subtraction of 2 nos is 10
multiplication of 2 nos is 75
division of 2 nos is 3

Web Page Design

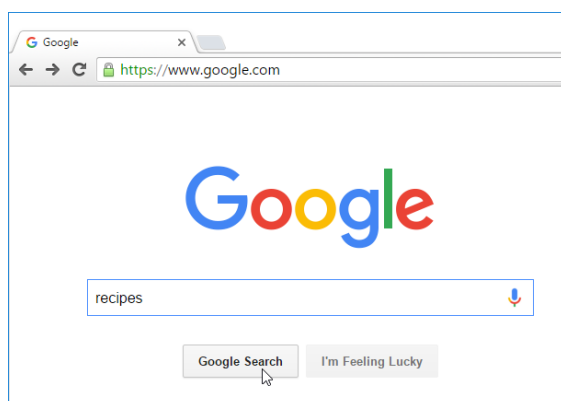
How to Browse different search engines and search different topics & create an email account & use attachment facility.

What is a Search Engine?

A search engine is a web-based tool that enables users to locate or search information on the World Wide Web on specific topics. Popular examples of search engines are Google, Bing, Yahoo!, and MSN Search etc.

How to search topics in the Web

To perform a search, we will need to navigate or browse to a search engine in our web browser. Type one or more **keywords**—also known as **search terms**—then press **Enter** on your keyboard. In this example, we'll search for **recipes**.



After you run a search, we'll see a list of **relevant websites** that match our search terms. These are commonly known as **search results**. If you see a site that looks interesting, you can click a link to open it.

CREATE AN E-MAIL ACCOUNT :

- Double click on internet explorer.
- Write www.yahoo.com or gmail.com or rediffmail.com in the address box then press enter.
- Click on create account .
- Then it will display a create account form.
- Filling the given form.
- Click on agree all form and condition.
- Your account will be created.

SENDING E-MAIL :

- Open the email website for. Ex. www.rediffmail.com or www.gmail.com
- Click in sign in , type user name and password.
- Click in write/compose option to write the mail
- Fill the information of TO, SUBJECT & type the message in the message box. For ex.
 To : bubu@gmail.com
 Sub : appointment letter
 Attach : appointment.doc (click in attachment button to attach any file)
- Then click in send button.

RECEIVING E-MAIL :

- Open the email website for. Ex. www.rediffmail.com or www.gmail.com
- Click in sign in , type user name and password.
- Click in inbox option to receive the mails
- Click on any message to open it.

HTML

HTML stands for Hypertext markup language. HTML is a method through which ordinary text can be converted into Hyptertext. It is a set of special code technically HTML is not a programming language. The language use to develop webpage is called HTML (Hyper text mark up language).

HTML is a language interpreted by a browser. Webpages are also called HTML document. HTML is a set of special code that can be embedded in to text to add formatting and linking information.

HTML ELEMENTS or TAGS

HTML is written in the form of tags. A tag refers to a string enclosed within angular brackets (< and >). HTML elements represent semantics or meaning.

- HTML tags are predefined and are not case-sensitive.
- All HTML tags are typed inside the angular brackets (< and >) and the text between these brackets are called elements.

- The opening tags are written within the less than (<) and greater than (>) signs,
e.g. <HTML>
- The closing tags are written within the < and > signs with a forward slash (/) appended before the name of the tag.
e.g. </HTML>
- The attributes are always specified in the opening tags, which provides additional information about that tag. It comes in name/value pairs like name = "value",
e.g.

There are two kinds of elements, i.e. Container elements and Empty elements.

Container Elements

These types of HTML elements always wrap around the text or graphics which come in a set with an opening as well as a closing tag. In other words, HTML tags that include both On and Off tags are known as container tags.

e.g.

<HTML>.. </HTML>

Empty Elements

On the other hand, the empty elements are standalone tags, i.e. empty tags do not have to be wrapped around text or graphics and do not require a closing tag. In other words, HTML tags that include only On tag and there is no Off tag then, these tags are known as empty tags.

e.g.

<HR> inserts a horizontal line

 adds a line break

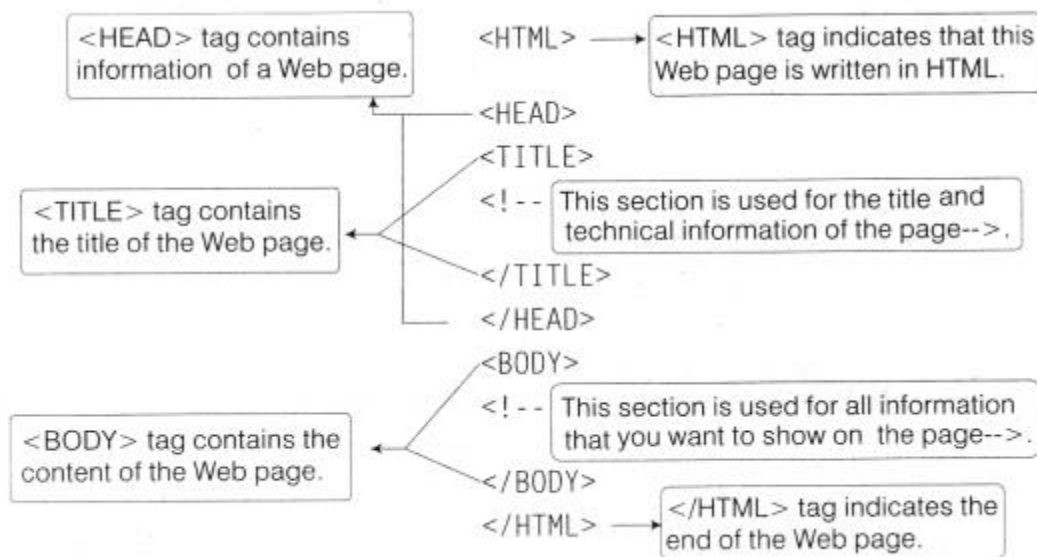
BASIC STRUCTURE OF HTML DOCUMENT

HTML document is a combination of various tags, which define the structure and appearance of the Web page.

Following four basic structure elements are always present in every HTML document:

- (i) The <HTML> tag tells the browser that this is an HTML document. You must begin your html files with this tag and must end your html file with matching closing tag.
- (ii) The <HEAD> tag is used for text and tags that do not appear directly on the page. It acts as a header of file and contains some information like setting the title of the Web page.
- (iii) The <TITLE> tag contains the document title. This tag lies between <HEAD> and </HEAD> tags. The title specified inside this tag appears on the browser's title bar.

(iv) The <BODY> tag is used for text and tags that appears directly on the Web page. It holds all your content words, pictures and other stuff. The most basic structure needed for creation of any Web page are as follows:



CREATING and SAVING a HTML DOCUMENT

Since, HTML documents are just text files, they can be written in any simple text editor like Notepad, FrontPage, Dreamweaver, WordPad etc.

In Notepad to create a HTML document, the steps are as follows:

- Step 1: Open Notepad Start —> search —> Notepad.
- Step 2: Type the HTML document in Notepad :

Examples :

```
<HTML>
<HEAD>
<TITLE> This is my first web page </TITLE>
</HEAD>
<BODY> welcome to my webpage </BODY>
</HTML>
```

- Step 3: Save the document (give the desired file name along with the extension .htm or .html in the File name then click Save button)

VIEWING HTML DOCUMENT in a BROWSER

You can view HTML document in any Web browser such as Internet Explorer, Mozilla Firefox, Opera, Netscape Navigator, Google Chrome etc. Here, we are using Internet Explorer Web browser to view the HTML document.

To view an HTML document, the steps are as follows:

- Step 1: Open Internet Explorer
- Step 2: To open the document First.html
- step 3: Now, you will be viewing your desired page

Different HTML TAGS

<CENTER> Tag

This tag is used to centralize a segment of text to be displayed on browser's window. With the <CENTER> tag, closing tag </CENTER> is always used. Anything between these two tags will be centered including text, images or tables.

Ex:

<CENTER>

This text will be center-aligned.

</CENTER>

Comment <!-- -> Tag

This tag is used to insert comments in the source code of the Web page. Comments are not displayed in the browser's window.

<!-- - This is a comment. - - >

Heading Tags

In HTML, heading tags are used to display the text as a heading. It can also be used to give section headings. There are six levels of headings, ranging from <H1>...</H1> to <H6>...</H6>. <H1> defines the most important largest heading level. <H6> defines the smallest heading level.

Syntax

<Hn>.....</Hn>

Where, n may be any number from 1 to 6.

Ex :

<BODY>

<H1> level 1 Head!ine </H1>

<H2> level 2 Headline </H2>

<H3> level 3 Headline </H3>

<H4> level 4 Headline </H4>

<H5> level 5 Headline </H5>

<H6> level 6 Headline </H6>

</BODY>

Paragraph <P> Tag

This tag is used to mark a block of text as a paragraph. It is used to insert a line break with extra space in the beginning. This is a container tag.

Syntax

<P>.....</P>

Ex :

```
<BODY>
<P>This is some text in a paragraph.</P>
</BODY>
```

Line Break
 Tag

This tag is used to insert a line break which means the text/image following the tag will be moved to the next line when displayed in the browser. To add a single line of space, you can use a break tag
. This is an empty tag, i.e. no need of closing tag. You can also use the
 tag to insert one or more blank lines.

Ex :

```
<BODY>
  This text contains <BR> a line break.
</BODY>
```

Horizontal Rule <HR> Tag

To create a horizontal line on your page, you have to use the empty tag <HR>. This horizontal line can be used to divide information into sections.

Attributes of <HR> Tag

Syntax

```
<HR align="left | center | right" size = "no">
```

Ex:

```
<BODY>
<HR align="right" size="3">
  This is a line.
</BODY>
```

Character Entities

Reserved characters in HTML must be replaced with character entities. Characters that are not present on our keyboard can also be replaced by entities. Some characters are reserved in HTML. If you use the less than (<) or greater than (>) signs in your text, the browser might mix them with tags. Character entities are used to display reserved characters in HTML.

A character entity looks like this :

&entity_name : OR &#entity_number;

| Result | Description | Entity Name | Entity number |
|--------|-----------------------|-------------|---------------|
| < | less than | < | &# 60; |
| > | greater than | > | > |
| & | ampersand | & | & |
| “ | double quotation mark | & quot; | " |

| | | | |
|---|-----------------------|----------|---------|
| ' | single quotation mark | & apos; | ' |
| £ | Pound | & pound; | £ |
| ¥ | Yen | & yen; | ¥ |
| € | Euro | & euro; | € |
| © | Copyright | & copy; | © |
| ® | Registered trademark | & reg ; | ® |

FORMATING ELEMENTS

| Tag | Description |
|----------|-----------------------------------|
| | bold text |
| | emphasized text |
| <i> | italic text |
| <u> | underlines |
| <small> | defines smaller text |
| <big> | defines bigger text |
| | defines important text |
| <sub> | defines subscripted text |
| <sup> | defines superscripted text |
| <ins> | defines inserted text |
| | defines deleted text |
| <mark> | defines marked / highlighted text |

MARQUEE TAG

It is used for scrolling piece of text or image displayed either horizontally across or vertically down your website page depending on the settings.

The <marquee> Tag Attributes

Attribute Description

Width : This specifies the width of the marquee. This can be a value like 10 or 20% etc

Height: This specifies the height of the marquee. This can be a value like 10 or 20% etc.

Direction : This specifies the direction in which marquee should scroll. This can be a value like *up*, *down*, *left* or *right*.

Behavior : This specifies the type of scrolling of the marquee. This can have a value like *scroll*, *slide* and *alternate*.

Scrolldelay : This specifies how long to delay between each jump. This will have a value like 10 etc.

Scrollamount : This specifies the speed of marquee text. This can have a value like 10 etc

Loop : This specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.

Bgcolor : This specifies background color in terms of color name or color hex value.

Hspace : This specifies horizontal space around the marquee. This can be a value like 10 or 20% etc

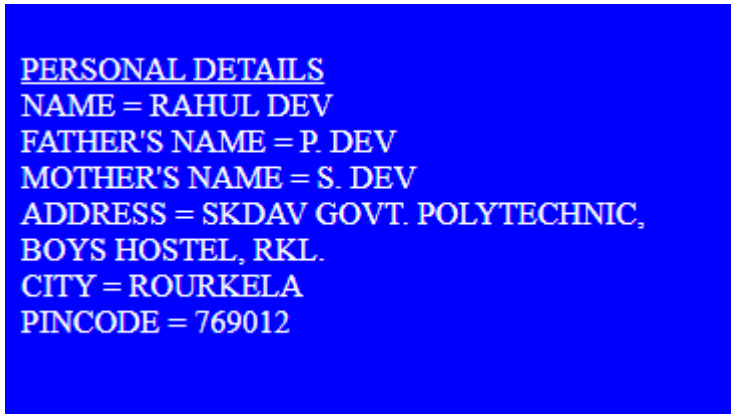
Vspace : This specifies vertical space around the marquee. This can be a value like 10 or 20% etc

Labwork-1

Design a webpage to print ur personal details by using background colour and break tag.

```
<HTML>
<HEAD>
<TITLE>Program – 1 </TITLE></HEAD>
<BODY BGCOLOR = "BLUE" TEXT = WHITE>
<BR> PERSONAL DETAILS
<BR> NAME = RAHUL DEV
<BR> FATHER'S NAME = P. DEV
<BR> MOTHER'S NAME = S. DEV
<BR> ADDRESS = SKDAV GOVT. POLYTECHNIC,
<BR> BOYS HOSTEL, RKL.
<BR> CITY = ROURKELA
<BR> PINCODE = 769012
</BODY>
</HTML>
```

output



PERSONAL DETAILS
NAME = RAHUL DEV
FATHER'S NAME = P. DEV
MOTHER'S NAME = S. DEV
ADDRESS = SKDAV GOVT. POLYTECHNIC,
BOYS HOSTEL, RKL.
CITY = ROURKELA
PINCODE = 769012

Labwork – 2

Design a web page to display “skdav” in different heading levels.

```
<HTML>
```

```

<HEAD><TITLE>Program - 2</TITLE></HEAD>
<BODY>
<H1> SKDAV </H1>
<H2> SKDAV </H2>
<H3> SKDAV </H3>
<H4> SKDAV </H4>
<H5> SKDAV </H5>
<H6> SKDAV </H6>
</BODY>
</HTML>

```

output

SKDAV

SKDAV

SKDAV

SKDAV

SKDAV

SKDAV

Labwork – 3

Design a webpage to display the text in different text styles / Text level tags like ****, **<i>**, **<u>**, **<small>****<big>**, **** etc.

```

<HTML>
<HEAD><TITLE>Program - 3</TITLE></HEAD>
<BODY BGCOLOR = PINK>
<BR> TEXT LEVEL TAG
<BR><BR><BR>
<B>Good Morning      </B>      <BR>
<U>Good Morning      </U>      <BR>
<I>Good Morning      </I>      <BR>
<SMALL>Good Morning </SMALL><BR>
<BIG>Good Morning    </BIG>    <BR>
<STRONG>Good Morning </STRONG><BR>
<EM> Good Morning    </EM>      <BR>
<MARK> Good Morning </MARK><BR>
<STRIKE> Good Morning </STRIKE><BR>
H<SUB>2</SUB>O          <BR>
A<SUP>2 </SUP> + B <SUP> 2 </SUP><BR>
<CENTER> Good morning </CENTER>
</BODY>
</HTML>

```


TEXT LEVEL TAG

Good Morning

Good Morning

Good Morning

Good Morning

Good Morning

Good Morning

Good Morning

Good Morning

~~Good Morning~~

H₂O

A² + B²

Good morning

Labwork -4

Design a web page to display the symbol of character entities.

```
<HTML>
```

```
<HEAD><TITLE> Program - 4 </TITLE></HEAD>
```

```
<BODY BGCOLOR = PINK>
```

```
<BR><CENTER> CHARACTER ENTITIES
```

```
<HR align="CENTER" WIDTH = 200 COLOR = GREEN>
```

```
<HR align="CENTER" WIDTH = 250 COLOR = RED>
```

```
<BR><BR>
```

| | | |
|-----------|-------|------|
| AMPERSANT | & |
 |
|-----------|-------|------|

| | | |
|------------|--------|------|
| COPY RIGHT | © |
 |
|------------|--------|------|

| | | |
|------------|---------|------|
| TRADE MARK | ™ |
 |
|------------|---------|------|

| | | |
|----------|-------|------|
| REGISTAR | ® |
 |
|----------|-------|------|

| | | |
|--------------|------|------|
| GREATER THAN | > |
 |
|--------------|------|------|

| | | |
|-----------|------|------|
| LESS THAN | < |
 |
|-----------|------|------|

| | | |
|----------|--------|------|
| JAPANESE | ¥ |
 |
|----------|--------|------|

| | | |
|--------------|--------|------|
| SINGLE QUOTE | ' |
 |
|--------------|--------|------|

| | | |
|--------------|--------|------|
| DOUBLE QUOTE | " |
 |
|--------------|--------|------|

```
</CENTER>
```

```
</BODY>
```

```
</HTML>
```

CHARACTER ENTITIES

AMPERSANT &
COPY RIGHT ©
TRADE MARK ™
REGISTAR ®
GREATER THAN >
LESS THAN <
JAPANESE ¥
SINGLE QUOTE '
DOUBLE QUOTE "

Labwork 5

Design a web page to print a paragraph by using background colour , font tag & marquee tag.

```
<HTML>
<HEAD><TITLE> PROGRAM – 5 </TITLE></HEAD>
<BODY BGCOLOR = pink>
<BIG><B><U><CENTER> Web Page Design </CENTER></BIG></B></U>
<BR>
<BR>
```

```
<FONT face = "tahoma" size = 5 color = red>
What is Web Design ? </font><BR>
```

```
<P align=justify><FONT face = "tahoma" size = 4 color = blue>
Web design refers to the design of websites that are displayed on the internet. It
usually refers to the user experience aspects of website development rather than
software development. Web design used to be focused on designing websites for
desktop browsers; however, since the mid-2010s, design for mobile and tablet
browsers has become ever-increasingly important. </font></P>
<BR>
<BR>
<FONT face = "arial black" size=3 color=PURPLE>
<MARQUEE DIRECTION = RIGHT BEHAVIOR = ALTERNATE > Welcome to
Web Page Design Lab class </MARQUEE></FONT>
<BR>
<BR>
```


What is web design and development?

<P align = justify>
Web design and development is an umbrella term that describes the process of creating a website. Like the name suggests, it involves two major skill sets: web design and web development. Web design determines the look and feel of a website, while web development determines how it functions. </P>

</BODY>
</HTML>

output



LINKS AND HYPERLINKS IN HTML

Link : A link is a chain that connects pages both within a website and to other websites. Without links, we wouldn't have any websites.

Hyperlink

A hyperlink, also called a link or web link, contains an address for a destination and acts as a reference to data. A user can easily follow, jump to, and be directed to the destination by either clicking, tapping on, or hovering over the link.

A hyperlink can be a piece of text, an image, an icon, or a graphic that, when you click on it, points to and navigates you to a different webpage or document. It can also point to a specific section or element within the same webpage or document.

Basically, hyperlinks are clickable pointers to a resource.

Hypertext is text with hyperlinks. The linked text (the reference to data) is called **anchor text**.

```
<a href="https://www.freecodecamp.org">Visit: Freecode Camp!</a>
```

LIST TAG

LIST Tag allow web developers to group a set of related items in lists.

There are three list types in HTML:

- **unordered list** — used to group a set of related items in no particular order
- **ordered list** — used to group a set of related items in a specific order
- **description list** — used to display name/value pairs such as terms and definitions

Unordered List

An unordered list starts with the `` tag. Each list item starts with the `` tag. The list items will be marked with bullets (small black circles) by default:

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Ordered List

An ordered list starts with the `` tag. Each list item starts with the `` tag. The list items will be marked with numbers by default:

Example

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Description Lists

A description list is a list of terms, with a description of each term.

The `<dl>` tag defines the description list, the `<dt>` tag defines the term (name), and the `<dd>` tag describes each term:

Example

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Labwork 6

Design a web page with background image, different text colour for different paragraphs, and set colors for links, active links and visited links.

```
<HTML>
<HEAD><TITLE> PROGRAM – 6 </TITLE></HEAD>
<BODY Background = "d:\logo.png" link=red alink=green vlink=pink>
<BIG><B><U><CENTER> HTML </CENTER></BIG></B></U>
<FONT face = "arial" size=3 color=blue>
```

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages. HTML is a method through which ordinary text can be converted into hypertext.

```
</FONT>
<BR><BR>
<B> HTML ELEMENTS or TAGS<B>
<BR>
<FONT face = "arial" size=3 color=green>
```

HTML is written in the form of tags. A tag refers to a string enclosed within angular brackets (< and >). HTML elements represent semantics or meaning.

```
<BR><BR></FONT>
```

```
<UL type = circle>
<LI>HTML tags are predefined and are not case-sensitive.
<LI>All HTML tags are typed inside the angular brackets (< and >) and the text
between these brackets are called elements.
</UL>
```

```
<OL>
<LI>The opening tags are written within the less than (<) and greater than (>)
signs,e.g. <HTML>
<LI>The closing tags are written within the < and > signs with a forward slash (/)
appended before the name of the tag.
e.g. </HTML>
```

The attributes are always specified in the opening tags, which provides additional information about that tag. It comes in name/value pairs like name = "value",

e.g.

 click here for search of 1st HTML document

click here for search of 2nd HTML document

</BODY>

</HTML>

HTML

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages. HTML is a method through which ordinary text can be converted into hypertext.

HTML ELEMENTS or TAGS

HTML is written in the form of tags. A tag refers to a string enclosed within angular brackets (< and >). HTML elements represent semantics or meaning.

- HTML tags are predefined and are not case-sensitive.
 - All HTML tags are typed inside the angular brackets (< and >) and the text between these brackets are called elements.
1. The opening tags are written within the less than (<) and greater than (>) signs, e.g.
 2. The closing tags are written within the < and > signs with a forward slash (/) appended before the name of the tag. e.g.
 3. The attributes are always specified in the opening tags, which provides additional information about that tag. It comes in name/value pairs like name = "value", e.g.

[click here for search of 1st HTML document](#)

[click here for search of 2nd HTML document](#)



TABLE TAG

A table is a two dimensional matrix consisting of rows & columns. In HTML pages table display data in a column format. The intersection of rows & columns is called a cell. A cell of a table in a page may contain simple text, Hypertext & image.

Table element tag

<table> tag – It holds others related tag required for creating of table such as rows , cells titles, alignments, borders etc.

<TR> tag – It is used to create a row of the table.

<TD> tag – It is table data tag. This pair of tag are used to specify the contains of a cell. This tag are used between the <TR> ...</TR> tag and the tags are required individually for each cell in the row.

A Table can have two types of row

- 1) Table Header Row (TH) – A table header row is defined using <TH>. The contains of a table header row is automatically control and appears in bold face.

Attribute :

Align – Align attribute can be set to left, right & centre.

Valign – It controls the vertical alignment of the cell contains. It accepts the value as top, bottom or moddle.

Border –It controls the border to the placed around the table.

Width – It sets the width to a specific no of pixel or to a % of the available screen with.

Cell Padding – It controls the distance between the data in the cell & the boundary of the cell.

Cell spacing – It controls the spacing between adjacent cells.

Colspan – This attribute is used inside a <TH> or <TD> tag. The colspan can be set equal to the no. of columns the cell is to occupy. This attribute is useful when one row of table needs to be a certain no. of column.

Rowspan – This attribute work same as the colspan except, that it allows a cell to take off more than one row.

Labwork 7

Create HTML table, format contents in table cells and span the rows and columns.

Ex:

```
<HTML>
```

```
<BODY>
```

```
<center>
```

```
<table border = 4 bordercolor=green bgcolor=yellow cellpadding = 5>
```

```
<tr>
```

```
<th> Month </th>
```

```
<th> Savings </th>
```

```
</tr>
```

```
<tr>
```

```

<td>January</td>
<td>5000</td>
</tr>
<tr>
<td>February</td>
<td>8000</td>
</tr>
<tr>
<td>March</td>
<td>6000</td>
</tr>
<tr>
<td colspan=2> Quarterly report  </td>
</tr>
<tr>
<td rowspan=2>April May</td>
<td>5500</td>
</tr>
<tr>
<td>6600</td>
</tr>
</table></center>
</BODY>
</HTML>

```

Output

| Month | Savings |
|------------------|---------|
| January | 5000 |
| February | 8000 |
| March | 6000 |
| Quarterly report | |
| April May | 5500 |
| | 6600 |

IMG TAG

HTML tag is used to add image inside webpage/website. Nowadays website does not directly add images to a web page, as the images are linked to web pages by using the tag which holds space for the image.

Syntax

``

Attributes: The `` tag has following attributes.

- **src:** It is used to specify the path to the image.
- **alt:** It is used to specify an alternate text for the image. It is useful as it informs the user about what the image means and also due to any network issue if the image cannot be displayed then this alternate text will be displayed.
- **height:** It is used to specify the height of the image.
- **width:** It is used to specify the width of the image.
- **sizes:** It is used to specify image sizes for different page layouts.

Ex: In this example we are using the `` tag along with `src`, `width`, `height`, and `alt` attributes.

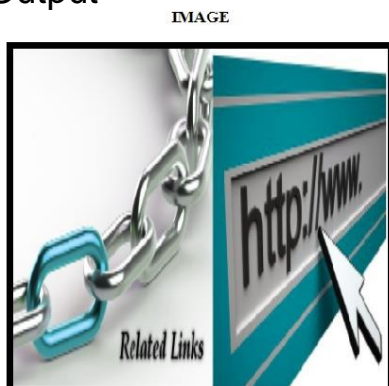
Labwork 8

Design a web page to insert an image.

```
<html>
<body style="text-align: center;">
<h3>IMAGE </h3>

</body>
</html>
```

Output



FRAMESET TAG

HTML `<frame>` tag define the particular area within an HTML file where another HTML web page can be displayed.

A <frame> tag is used with <frameset>, and it divides a webpage into multiple sections or frames, and each frame can contain different web pages.

Syntax

```
<frameset cols = " %,  %">
<frame src = "URL">
<frame src = "URL">
</frameset>
```

Attributes

- Frameborder (0 or 1) -It specifies whether to display a border around the frame or not, and its default value is 1
- Marginheight - It specifies the top and bottom margins of the frame.
- Marginwidth - It defines the height of the margin between frames.
- Name - It is used to assign the name to the frame.
- Noresize - It is used to prevent resizing of the frame by the user.
- Scrolling (yes / no / auto) - It specifies the existence of the scrollbar for overflowing content.
- Src - It specifies the URL of the document which we want to display in a frame.

Labwork 9

Create basic frameset and format the frames within the frameset using different attributes. Also use frame targeting.

Ex :

Main.html

```
<html>
<head>
<title>Frame tag</title>
</head>
<frameset cols="25%,50%,25%">
<frame src="d:\frame1.html">
<frame src="d:\frame2.html">
<frame src="d:\frame3.html">
</frameset>
</html>
```

Frame1.html

```
<html>
<body bgcolor=green>
<Br> <center>
<h2>This is first frame</h2> </center>
```

```
</body>
</html>
```

Frame2.html

```
<html>
<body bgcolor=yellow>
<br><center>
<h2>This is Second frame</h2> </center>
</body>
</html>
```

Frame3.html

```
<html>
<body bgcolor = pink>
<br><center>
    <h2>This is Third frame</h2> </center>
</body>
</html>
```

Output



FORM TAG

An HTML form is a section of a document which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter / input data that is to be sent to the server for processing such as name, email address, password, phone number, etc.

Syntax

```
<form action="server url" method="get|post">
```

//input controls e.g. textfield, textarea, radiobutton, button

```
</form>
```

HTML Form Tags

| Tag | Description |
|-----|-------------|
|-----|-------------|

| | |
|------------|--|
| <form> | It defines an HTML form to enter inputs by the used side. |
| <input> | It defines an input control. |
| <textarea> | It defines a multi-line input control. |
| <label> | It defines a label for an input element. |
| <fieldset> | It groups the related element in a form. |
| <legend> | It defines a caption for a <fieldset> element. |
| <select> | It defines a drop-down list. |
| <optgroup> | It defines a group of related options in a drop-down list. |
| <option> | It defines an option in a drop-down list. |
| <button> | It defines a clickable button. |

Labwork 10

Create basic form using different input controls and pulldown menu.

<HTML>

<body>

<form>

Enter your name
 <input type="text" name="username">

First Name: <input type="text" name="firstname"/>

Last Name: <input type="text" name="lastname"/>

Enter your address:
 <textarea rows="2" cols="20"></textarea>

Gender <input type = "radio" name = gender value = "male"> MALE

<input type = "radio" name = gender value = "female"> FEMALE

Enter email id <input type="email" name ="email">

Enter hobbies <input type = "checkbox" valued = reading> READING

<input type = "checkbox" valued = writing> WRITTING

<input type = "checkbox" valued = singing> SINGING

<input type = "checkbox" valued = dancing> DANCING

Date of Birth

<select name = "date">

<option> 1

<option> 2

<option> 3

<option>4

</select>

<select name = "month">

<option> jan

<option> feb

```

        <option> mar
        <option> april
    </select>
    <select name = "year">
        <option> 1990
        <option> 1991
        <option> 1992
        <option> 1993
    </select><br><br>

    City
    <select name = "city">
        <option value = "Rourkela"> RKL </option>
        <option value = "BBSR"> BBSR </option>
        <option value = "Cuttack"> CTC </option>
        <option value = "Baleswar"> BSWR </option>
    </select>

    <input type = "submit" value = "SUBMIT">
    <input type = "submit" value = "RESET">

    </form>
</body>
</HTML>

```

FILL THE PERSONAL DETAIL FORM

First Name :

Last Name :

Address :

Gender MALE FEMALE

Enter email id

Enter hobbies ☐ READING

writing WRITING

singing SINGING

dancing DANCING

Date of Birth 1 jan 1990

City

Labwork 11

Design web page and include different lists & various links in a web page & include images with different alignments and wrapped text in web page. Also include image as a link in the web page.

<!Doctype Html>

```

<Html>
<Head>
<Title>
Wrap the text
</Title>
<style>
div {
    width: 400px;
    border: 2px solid red;
}
div.a {
    word-wrap: normal;
}
div.b {
    word-wrap: break-word;
}
</style>
</Head>
<Body bgcolor="yellow">
<center><h1><u> MULTIMEDIA AUTHORISING TOOLS </u></h1></center>
<pre> It is also known as author ware or programs that helps us to write
hypertext or multimedia application.
This tools usually enables to create a final application by linking together objects,
like paragraph of a text,
an illustration or sum. </pre>

```

**DIFFERENT AUTHORIZING TOOLS **

```

<UL type = circle>
<LI>Multimedia authoring way
<LI>Multimedia director
<LI>Multimedia flash
</UL>

```

```

<a href = "D:\dd3.html"> Click Here </a> for visit next page
<br><br>
<img scr = "file:/D:/LOTUS.JPG" ALIGN="right" width=50 height = 50 border=2>
<br>
<a href = d:\dd4.html>

click the image
</a>

```

<center> F L A S H

<div class="a">

Flash is an interactive multimedia and web authoring software. It has become the professional standard for multimedia and web animation. It is not only used to design complete interactive web sites but also arcade style games and database applications. Flash animations are faster than animated GIFs because

</div>

<div class="b">

Flash uses vector graphics instead of raster graphics and the file size is very small. Flash was created by Jonathan Gay, an employee of Silicon Beach software Company.

</div> </center>

</Body>

</Html>


MULTIMEDIA AUTHORISING TOOLS

It is also known as author ware or programs that helps us to write hypertext or multimedia application. This tools usually enables to create a final application by linking together objects, like paragraph of a text, an illustration or sum.

DIFFERENT AUTHORIZING TOOLS

- o Multimedia authoring way
- o Multimedia director
- o Multimedia flash

[Click Here](#) for visit next page

 [click the image](#)

FLASH

Flash is an interactive multimedia and web authoring software. It has become the professional standard for multimedia and web animation. It is not only used to design complete interactive web sites but also arcade style games and database applications. Flash animations are faster than animated GIFs because

Flash uses vector graphics instead of raster graphics and the file size is very small. Flash was created by Jonathan Gay, an employee of Silicon Beach software Company.

HTML Multimedia

Multimedia

Multimedia represents various types of media content, used together. If we add a text with pictures, add a video, audio, image then we have multimedia. Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more.

Video clips

Today, embedding a video clip in a [web page](#) is more than easy, and video clips can be easily shared via our blog, forum or even a small video commercial which we can put on our online store.

Audio

Audio is another thing, which was affected by the technology advances. And while a video clip will require much more bandwidth and server resources, audio files are much faster to process from a server's point of view. This helped for their distribution among website creators much faster. Today, a website with a musical background is something more than normal, and on a lot of websites you can easily listen to an audio track, which is uploaded on the web server with the help of a simple player.

Images

Images were the first type of multimedia to be ever used in a web page. They quickly became an indispensable part of the website design process. Images became also an important part of the layout of a web page - images could now be used for buttons, navigational menus and backgrounds.

Browser Support

The first web browsers had support for text only, limited to a single font in a single color. Later came browsers with support for colors, fonts, images, and multimedia!

Multimedia Formats

Multimedia elements (like audio or video) are stored in media files.

The most common way to discover the type of a file, is to look at the file extension.

Multimedia files have formats and different extensions like: .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

Common Video Formats

| Format | File | Description |
|--------|---------------|---|
| MPEG | .mpg
.mpeg | MPEG. Developed by the Moving Pictures Expert Group. The first popular video format on the web. Not supported anymore in HTML. |
| AVI | .avi | AVI (Audio Video Interleave). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers. |
| WMV | .wmv | WMV (Windows Media Video). Developed by Microsoft. Commonly used in video cameras and TV hardware. Plays well on Windows computers, but not in web browsers. |
| Flash | .swf | Flash. Developed by Macromedia. Often requires an extra |

| | | |
|---------------|-------|---|
| | .flv | component (plug-in) to play in web browsers. |
| WebM | .webm | WebM. Developed by Mozilla, Opera, Adobe, and Google. Supported by HTML. |
| MPEG-4 or MP4 | .mp4 | MP4. Developed by the Moving Pictures Expert Group. Commonly used in video cameras and TV hardware. Supported by all browsers and recommended by YouTube. |

Common Audio Formats

MP3 is the best format for compressed recorded music. The term MP3 has become synonymous with digital music.

| Format | File | Description |
|--------|------|--|
| MP3 | .mp3 | MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers. |
| MP4 | .mp4 | MP4 is a video format, but can also be used for audio. Supported by all browsers. |

HTML Video

The HTML `<video>` element is used to show a video on a web page.

Attributes of Video tag / element

The `controls` attribute adds video controls, like play, pause, and volume.

It is a good idea to always include `width` and `height` attributes. If height and width are not set, the page might flicker while the video loads.

The `<source>` element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.

The text between the `<video>` and `</video>` tags will only be displayed in browsers that do not support the `<video>` element.

Autoplay – To start a video automatically.

Add `muted` after `autoplay` to let your video start playing automatically (but muted)

Ex1

```
<html>
<body>
welcome to my multimedia page <br>
<br>
This is the 1st video <br><br>
<video width="500" height="500" controls>
  <source src="e:\sample-mp4-file-small.mp4" type="video/mp4">
```

Your browser does not support the video tag.

```
</video>
</body>
</html>
```

Ex2

```
<html>
<body>
welcome to my multimedia page <br>
<br>
This is the 1st video <br><br>
<video width="500" height="500" autoplay muted>
  <source src="e:\sample-mp4-file-small.mp4" type="video/mp4">
```

Your browser does not support the video tag.

```
</video>
</body>
</html>
```

HTML Audio

The HTML **<audio>** element is used to play an audio file on a web page.

Attributes of Audio tag / elements

The **controls** attribute adds audio controls, like play, pause, and volume.

The **<source>** element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

The text between the **<audio>** and **</audio>** tags will only be displayed in browsers that do not support the **<audio>** element.

Autoplay - To start an audio file automatically

Ex1

```
<html>
<body>
welcome to my multimedia page <br>
<br>
This is the 1st audio <br><br>
<audio controls autoplay>
  <source src="e:\bharat.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
</body>
</html>
```

Ex2

```
<html>
<body>
welcome to my multimedia page <br>
<br>
This is the 2nd audio <br><br>
<audio controls autoplay muted>
  <source src="e:\bharat.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
</body>
</html>
```

Labwork 12

How to upload a Webpage on internet.

Step 1 :- Create a New Folder in any drive(D: or E:)

Step 2:- Design the HTML Web page by using HTML tag.

Step 3:- Save the webpage inside the New Folder as the name index.html.

Step 3:- Open the website <https://www.netlify.com/>

Step 4:- Click on Signup -> Email (Create a netlify account.)

Step 5:- Write your Gmail id and password in the required fields.

Step 6:- Verify the mail by opening the mail id. -> Fill-up the form and click on Setup and Continue.

Step 7:-Login in it.

Step 8 :- click in browse to upload(select the folder to upload)-upload-> click in domain setting- custom domain->option-> edit site name-> write down your suitable site name(soumyaranjan)-> click in save.

Step 9 : open the browser (google)-> write down the site name in address bar for ex. (soumyaranjan.netlify.app)

Now the webpage is display.