



Character's

Character's	ASCII Range
0-9	48-57
A-Z	65-90
a-z	97-122
Other character's	0-255 excluding the above.

Switch STATEMENT.

SYNTAX:-

```

switch (exp)
{
  case constant-1 : st-1;
                    break;
  case constant-2 : st-2;
                    break;
                    :
                    :
                    :
  default : st-n;
}
    
```

when \* switch, case, & default are keywords.

\* break - break statement is a statement which is used to send the control out of switch statement.

\* exp - expression is an integer or character wise like:

- \* constant - 1, 2, ..., n are any integer constant or character's constant.
- \* statement - are executable part which executes when exp. matches with constant.

SYNTAX: while (condition) statement;

WHILE  
 when "condition" is any boolean expression whose result may be true or false & statement may be simple or compound statement.

DO-While STATEMENT:-

In this statement, first of all, we execute the loop statement & then condition is checked. \* In do-while statement, if condition is false for the very first time even then the statements will execute once.

```

SYNTAX:- do { st;
           } while (condition);
    
```

FOR STATEMENT

FOR statement is used to initialise the counter variable, check the condition & also increment or decrement the counter variable & execute the loop statement until the condition is true.

```

SYNTAX:- for (exp-1; exp-2; exp-3)
           statement;
    
```

NESTED FOR STATEMENT.

```

SYNTAX:- for (exp-1; exp-2; exp-3)
           {
             for (exp-4; exp-5; exp-6)
               statement;
           }
    
```

goto statement

```

SYNTAX:- goto (row column);
    
```

Ex: int i=1;  
 A: cout << "sum" << endl; goto A;  
 if (i <= 10) i=i+1;  
 goto A;

break STATEMENT:- SYNTAX:- break;



STANDARD LIBRARY FUNCTIONS. \* use <string.h> header file.

STRING HANDLING FUNCTIONS.

(1). strlen (s); → This function is used to calc. the length of string.

Syntax: - n = strlen (st);

st → string variable or constant.  
n → integer variable.

```
Ex:- char s[] = "Welcome";
int n = strlen (s);
cout << n; → 7.
n = strlen ("C++ Subject")
cout << n; → 11
```

(2). strcpy (s); → used to copy the string into string variable.

Syntax: - strcpy (st1, st2);

st1, st2 → string var. or constant.

\* The value of st2 will be copied into var. st1.

```
Eg:- char s1[] = "C++";
strcpy (s1, s2);
cout << s1; → C++
```

```
strcpy (s1, "Hello");
cout << s1; → Hello.
```

(3). strcmp (s1, s2) → used to compare 2 string. If both the strings are equal it will return zero, if the first string is greater, it will return +ve value (+1) otherwise it will return -ve value (-1).

Syntax: - int <n> = strcmp (st1, st2);

```
Eg:- int n = strcmp ("Abc", "abc");
if (n == 0) cout << "Both strings are equal";
else cout << " " " not " " "
```

of if (strcmp ("Abc", "abc") == 0)

cout << "equal";

else cout << "Not equal";

(4). strcat (s1, s2) → used to concatenate two strings.

Syntax: - strcat (s1, s2);

\* value of st2 is concatenated by value of st1 & stored in variable st1;

Eg:-

```
char n[] = "C++", m[] = "2021";
strcpy (n, "Computer");
strcpy (m, "Science");
strcat (n, m);
cout << n; → Computer Science
cout << m; → Science.
```

CHARACTER FUNCTIONS. \* use <ctype.h> header file.

(1). islower (c)

Syntax: - int n = islower (ch);  
This func's returns 1 if char. is in lower case otherwise 0.

(2). isupper (c)

Syntax: - n = isupper (ch);

(3). isalpha (c) → alphabet or not.

Syntax: - n = isalpha (ch);

(4). isdigit (c) → digit or not

Syntax: - n = isdigit (ch);

(5). toupper (c) → convert its argument into uppercase if it is in lower case.

Syntax: - ch1 = toupper (ch2);

(6). tolower ( )

```
ch1 = tolower (ch2);
```

\* CONSOLE INPUT-OUTPUT FUNCTIONS, use <stdio.h>

(i). getchar ( ) used to read a character from the keyboard & store it in variable.

```
SYNTAX:- var = getchar ( );
```

```
Eg:- char ch;
      ch = getchar ( );
      cout << ch;
```

(ii). putchar ( ) used to display a single character on the screen.

```
SYNTAX:- putchar (ch);
```

```
Eg:- char ch = 'a';
      putchar (ch); → a
      putchar ('A'); → A.
```

(iii). gets ( ) used to input a string of characters.

```
SYNTAX:- gets (st);
```

(iv). puts ( ) used to display the string value on screen. It ends with a newline character.

```
SYNTAX:- puts (st);
```

\* INPUT-OUTPUT STREAM FUNCTION. use <iostream.h>

(i). get ( ) → used to input a single character.

```
SYNTAX:- cin.get (ch);
```

(ii). put ( ) → used to display a single character.

```
SYNTAX:- cout.put (ch);
```

(iii). getline ( ) → used to input a string with spaces & it ends with a newline character.

```
SYNTAX:- cin.getline (st, size);
```

```
Eg:- show n [20];
      cin.getline (n, 20);
```

(iv). write ( ) → used to display the string.

```
SYNTAX:- cout.write (st, size);
```

```
Eg:- cout.write ("welcome", 4);
```

\* STRUCTURES.

```
SYNTAX:- struct structure name
```

```
{ number 1;
  number 2;
  ;
};
```

(i). Eg:- struct student

```
{ int r;
  char n [20];
  float p;
```

→ To access the members of a structure, we req. a structure variable.

```
student s1, s2, s3;
```

(ii).

```
struct
```

```
{ int r;
  char n [20];
  float p;
  } s;
```



Q. W.A.P to input even nos. upto an inputted no.

```
#include <iostream.h>
void main ( )
{
  class c;
  int n, i;
  cout << "Enter the no. ";
  cin >> n;
  i = 2;
  while (i <= n)
  {
    cout << i << endl;
    i = i + 2;
  }
  getch ( );
}
```

Q. W.A.P to input n random nos. & find out the sum of those random nos.

```
#include <iostream.h>
void main ( )
{
  class c;
  int n, a, s;
  s = 0;
  cout << "Enter the no. ";
  cin >> n;
  i = 1;
  while (i <= n)
  {
    cout << "Enter any no. ";
    cin >> a;
    s = s + a;
    i++;
  }
  cout << "Sum = " << s;
  getch ( );
}
```

Q. W.A.P to input any no. & reverse that no. {palindrome}

```
#include <iostream.h>
void main ( )
{
  class c;
  int n, r = 0, a;
  cout << "Enter the no. ";
  cin >> n;
  m = n;
  while (n > 0)
  {
    a = n % 10;
    n = n / 10;
    r = r * 10 + a;
  }
  if (r == m)
  cout << "Palindrome ";
}
```

Q. W.A.P to input any no. & check whether the no. is palindrome.

```
#include <iostream.h>
void main ( )
{
  class c;
  int n, m, r = 0;
  cout << "Enter any no. ";
  cin >> n;
  m = n;
  while (n > 0)
  {
    a = n % 10;
    n = n / 10;
    r = r * 10 + a;
  }
  if (r == m)
  cout << "Palindrome ";
}
```

Q. W.A.P to input any no. & reverse that no. {palindrome}

```
#include <iostream.h>
void main ( )
{
  class c;
  int n, r = 0, a;
  cout << "Enter the no. ";
  cin >> n;
  while (n > 0)
  {
    a = n % 10;
    n = n / 10;
    r = r * 10 + a;
  }
  if (r == n)
  cout << "Reverse no. = " << r;
  getch ( );
}
```

Q. W.A.P to input any no. & check whether the no. is palindrome.

```
#include <iostream.h>
void main ( )
{
  class c;
  int n, m, r = 0;
  cout << "Enter any no. ";
  cin >> n;
  m = n;
  while (n > 0)
  {
    a = n % 10;
    n = n / 10;
    r = r * 10 + a;
  }
  if (r == m)
  cout << "Palindrome ";
}
```

```

else cout << "Not a palindrome";
    getch();
}

```

Q. WAP to input any no. & find out the sum of individual digits of that no. Ex: → if n is 723 then output be 12.  $7+2+3$

```

#include <iostream.h>
using namespace std;
void main()
{
    char c;
    int n, s=0;
    cin >> n;
    while (n > 0)
    {
        a = n % 10;
        n = n / 10;
        s = s + a;
    }
    cout << "Sum = " << s;
    getch();
}

```

Q. WAP to check whether the inputted no. is prime or not.

```

#include <iostream.h>
using namespace std;
void main()
{
    char c;
    int n, i, m=0, flag=0;
    cout << "Enter the No. to check prime";
    cin >> n;
    m = n / 2;
    for (i=2; i <= m; i++)
    {
        if (n % i == 0)
        {
            flag = 1;
            break;
        }
    }
    if (flag == 0)
        cout << "No. is prime";
    else
        cout << "No. is not prime";
}

```

```

if (n % i == 0)
{
    cout << "No. is not prime" << endl;
    flag = 1;
    break;
}
}
if (flag == 0)
    cout << "No. is prime" << endl;
getch();
}

```

Q. WAP to find out the area of square, rectangle & circle depending upon the user choice.

```

#include <iostream.h>
using namespace std;
void main()
{
    char c;
    int a, b;
    float r;
    char ch;
    do {
        char c;
        cout << "\t\t Main Menu\n";
        cout << "\t\t 1 for square area\n";
        cout << "\t\t 2 for rectangle area\n";
        cout << "\t\t 3 for circle area\n";
        cout << "\t\t 4 for quit\n";
        cout << "Enter your choice";
        cin >> ch;
        switch (ch)
        {
            case '1':
                cout << "Enter the side";
                cin >> a;
                cout << "Area of square = " << a * a << endl;
            case '2':
                cout << "Enter length & breadth";
                cin >> a;
                cin >> b;
                cout << "Area of rectangle = " << a * b << endl;
            case '3':
                cout << "Enter radius";
                cin >> r;
                cout << "Area of circle = " << 3.14 * r * r << endl;
            case '4':
                cout << "Program terminated" << endl;
                break;
            default:
                cout << "Invalid choice" << endl;
        }
        cout << "Do you want to continue (y/n)";
        cin >> ch;
    } while (ch == 'y');
}

```

```

do {
    char c;
    cout << "\t\t Main Menu\n";
    cout << "\t\t 1 for square area\n";
    cout << "\t\t 2 for rectangle area\n";
    cout << "\t\t 3 for circle area\n";
    cout << "\t\t 4 for quit\n";
    cout << "Enter your choice";
    cin >> ch;
    switch (ch)
    {
        case '1':
            cout << "Enter the side";
            cin >> a;
            cout << "Area of square = " << a * a << endl;
        case '2':
            cout << "Enter length & breadth";
            cin >> a;
            cin >> b;
            cout << "Area of rectangle = " << a * b << endl;
        case '3':
            cout << "Enter radius";
            cin >> r;
            cout << "Area of circle = " << 3.14 * r * r << endl;
        case '4':
            cout << "Program terminated" << endl;
            break;
        default:
            cout << "Invalid choice" << endl;
    }
    cout << "Do you want to continue (y/n)";
    cin >> ch;
} while (ch == 'y');
}

```



```

r = a * a;
cout << "Area of square = " << r;
break;

```

```

case 'r': cout << "Enter the length and then breadth";
cin >> a >> b;
r = a * b;
cout << "area of rectangle = " << r;
break;

```

```

case 'c': cout << "Enter the radius";
cin >> r;
r = 3.14 * a * a;
cout << "area of circle = " << r;
break;

```

```

case 'q':
case 'Q': cout << "Thank You";
break;
default : cout << "Wrong choice";
}

```

```

getch();
while (ch != 'Q' && ch != 'q');
}

```

Q. W.A.P to input any no. & find out the factorial

```

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

```

```

class c;
int n, i, f=1;
cout << "Enter the no.";
cin >> n;

```

```

for (i=1; i<=n; i++)
{ f = f * i; }
cout << "Factorial = " << f;

```

```

getch();
}

```

Q. W.A.P to display n terms of Fibonacci series;

```

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

```

```

class c;
int n, i, a=0, b=1, c;

```

```

cout << "Enter any no.";
cin >> n;

```

```

cout << a << endl;
cout << b << endl;

```

```

for (i=3; i<=n; i++)
{ c = a + b;
cout << c << endl;

```

```

a = b;
b = c;
}

```

```

getch();
}

```

011235.....

Q. WAP to find out the sum of following series:-

```

x/2 + x/3 + x/4 + ... + x/n
#include <iostream.h>
#include <conio.h>
#include <math.h>
void main ()
{ clrscr ();
  int x, n, i;
  float s=0;
  float t;
  cout << "Enter the value x & n";
  cin >> x >> n;
  for (i=2; i<=n; i++)
  { t = x / pow (i, 2);
    s = s + t;
  }
  cout << "Sum of series = " << s;
  getch ();
}

```

Q. W.A.P to find out the sum of following series:-

```

x/2 + x^2/3 + x^3/4 + ... + x^n/n+1
void main ()
{ clrscr ();
  int x, n, i;
  float s=0, t;
  cout << "Enter x & n";
  cin >> x >> n;
}

```

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

```

{ t = pow(x, i-1) / i;
  s = s + t;
}
cout << "Sum = " << s;
getch ();
}

```

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

```

{ t = pow(x, i) / i+1;
  s = s + t;
}

```

Q. WAP to find the sum of following series:

$$x/2 - x^2/3 + x^3/4 - \dots + \frac{x^n}{n+1}$$

only logic

```

void main ()
{ int x, n, i;
  float t, s=0;
  int p=1;
}

```

```

cout << "Enter x & n";
cin >> x >> n;
for (i=1; i<=n; i++)
{ t = pow (x, i) / i+1;
  t = t * p;
}

```

if (p==1) p=-1;

else p=1;

```

s = s + t;
cout << "sum = " << s;
getch ();
}

```



Q. W.A.P to find out the sum of following series:-

$$\frac{1}{a_1} + \frac{1}{3i} + \frac{1}{a_1} + \dots + \frac{1}{n}$$

```
void main ()
{
    class C ();
    int n, i, j, f, s=0, t;
    cout << "Enter n";
    cin >> n;
    for (i=2; i<=n; ++i)
    {
        f=1;
        for (j=1; j<=i; ++j)
        {
            f = f * j;
            t = 1/f;
            s = s + t;
        }
        cout << "Sum = " << s;
        getch();
    }
}
```

Q. W.A.P to find out the sum of series:-

$$(2^2) + (2^2 + 4^2) + (2^2 + 4^2 + 6^2) + \dots + (2^2 + \dots + n^2)$$

```
void main ()
{
    class C ();
    int n, i, j, s=0, f, t;
    cout << "Enter n";
    cin >> n;
    for (i=1; i<=n; ++i)
    {
        f=2; t=0;
        for (j=2; j<=i; ++j)
        {
            t = t + f;
            f = f + 2;
        }
        s = s + t;
    }
    cout << "Sum = " << s;
    getch();
}
```

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

```
{
    a = f * t;
    t = t + a;
    f = f + 2;
}
```

```
cout << "sum = " << s;
getch();
```

Q. W.A.P to input the value of n & display the output as shown. Ex: If n=5, output will be

```
* * *
* * * *
* * * * *
* * * * *
* * * * *
```

```
void main ()
{
    class C ();
    int n, i, j;
    cout << "Enter n ";
    cin >> n;
    for (i=1; i<=n; ++i)
    {
        for (j=1; j<=i; ++j)
        {
            cout << " * ";
        }
        cout << endl;
    }
}
```

getch();

output :-

```
* * * * *
* * * *
* * *
* *
*
```

Logic: → for (i=n; i>=1; --i) { for (j=n; j>=i; --j) { cout << " \* "; } cout << endl; }





Q result :->

```

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

```

```

int i, j, J, K, n;
char ch = 'A';
cout << "Enter n ";
cin >> n;

for (i=1; i<=n; ++i)
{
  for (j=n; j>=i; j--)
  {
    cout << "-";
    ch = ch + i;
    for (K=i; K>=1; --K)
    {
      cout << "--ch;
    }
    for (J=2; J<=i; ++J)
    {
      cout << ++ch;
    }
    ch = 'A';
    cout << endl;
  }
  getch();
}

```

Q. WAP to input any password, check & display whether the password is correct or not. If the password is INDIA.

```
#include <iostream.h>
#include <conio.h>
#include <string.h>
void main ()
{ clrscr();
  char a[20];
  cout << "Enter the Password";
  cin >> a;
  if (strcmp (a, "INDIA") == 0)
    cout << "Password is correct";
  else
    cout << "Incorrect";
  getch();
}
```

Q. What will be the output of given segment :-

```
char s[] = "Computer_Sc.2017";
for (int i=0; s[i] != '\0'; ++i)
{ if (!isalpha(s[i]))
  s[i] = '@';
  else if (isupper(s[i]))
    s[i] = s[i++];
  else s[i] = s[i] + 1; }
cout << s;
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C	o	m	P	U	T	e	R	-	S	c	.	2	0	1	7

output: → o p n U T e f - @ c d @ @ @ @ @

∴ output → opnUTef-@cd@@@@



Q. WAP to enter a line. Count & display total no. of words in that line.

```
#include <iostream.h>
#include <conio.h>
using namespace std;
void main()
{
```

```
char a[100];
```

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

```
cout << "Enter any line";
```

```
gets(a);
```

```
for (i=0; a[i]!='\0'; i++)
```

```
if (a[i] == ' ')
```

```
++j;
```

```
cout << "No. of words = " << j;
```

```
getch();
```

```
}
```

WAP to input to elements in an array and display them.

```
#include <iostream.h>
```

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

```
void main()
{
```

```
int a[10], i;
```

```
for (i=0; i<=9; i++)
```

```
{ cout << "Enter the elements";
```

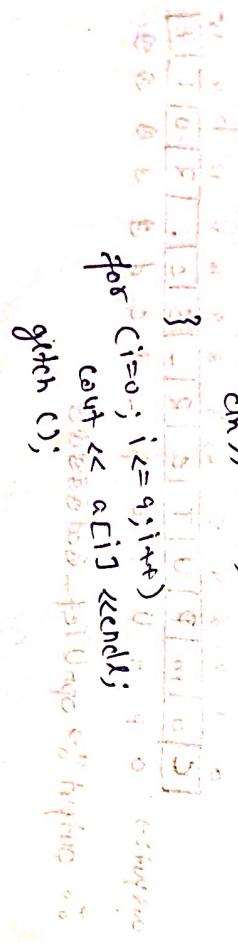
```
cin >> a[i];
```

```
}
```

```
for (i=0; i<=9; i++)
```

```
cout << a[i] << endl;
```

```
getch();
```



Q. W.A.P to input the sum of even elements in an array & find out

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

```
int a[10], i, s=0;
```

```
for (i=0; i<=9; i++)
```

```
{ cout << "Enter no.:";
```

```
cin >> a[i];
```

```
}
```

```
for (i=0; i<10; i++)
```

```
{ if (a[i] % 2 == 0)
```

```
{ s = s + a[i];
```

```
cout << "sum = " << s;
```

```
getch();
```

```
}
```

Q. WAP to input no. in an array and find out the largest no. in the array.

```
#include <iostream.h>
```

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

```
void main()
{
```

```
int i, a[10], l;
```

```
for (i=0; i<=9; i++)
```

```
{ cout << "Enter no.:";
```

```
cin >> a[i];
```

```
}
```

```
l = a[0];
```

```
for (i=0; i<=9; i++)
```

```
{ if (a[i] < a[l])
```

```
l = a[i];
```

```
}
```

```
cout << "largest No. = " << l;
```

```
getch();
```

```
}
```

Q. WAP to input to nos. in an array & search for a no. whether it is present or not in the array.

```

#include <iostream.h>
#include <conio.h>
void main ()
{
  int a[10], i, f=0, n;
  for (i=0; i<9; ++i)
  {
    cout << "Enter to nos. ";
    cin >> a[i];
  }
  cout << "Enter no. to be searched "; (cin >> n);
  for (i=0; i<9; ++i)
  {
    if (a[i] == n)
    {
      cout << "No. is found";
      f = 1;
    }
  }
  if (f == 0)
  cout << "No. is not found";
  getch();
}

```

Q. WAP to input an array & display the array & reversed if that array.

```

#include <iostream.h>
#include <conio.h>
void main ()
{
  int a[10], i, m, n, t;
  cout << "Enter an array";
  for (i=0; i<9; ++i)
  {
    cin >> a[i];
  }
  cout << "Enter an array";
  for (i=0; i<9; ++i)
  {
    cout << a[i];
  }
  t = n-1;
  for (i=0; i<9; ++i)
  {
    t = a[i];
    a[i] = a[t];
    a[t] = t;
  }
  getch();
}

```

```

}
cout << "Reversed array";
for (i=0; i<9; ++i)
{
  cout << a[i];
}
getch();
}

```

Q. WAP to input an array & interchange first half array element with second half.

```

void main ()
{
  int i, j, n, a[20], t;
  cout << "Enter size of array not more than 20";
  cin >> n;
  for (i=0; i<n; ++i)
  {
    cout << "Enter array";
    cin >> a[i];
  }
  t = n/2;
  for (i=0; i<n/2; ++i)
  {
    t = a[i];
    a[i] = a[t];
    a[t] = t;
  }
  cout << "Reversed array";
  for (i=0; i<n; ++i)
  {
    cout << a[i];
  }
  getch();
}

```



Q. WAP to input an array & sort them in ascending order:-

```

void main ( )
{ int a [20], n, i, j, t;
  cout << "Enter size";
  cin >> n;
  for (i=0; i < n; i++)
  { cout << "Enter array";
    cin >> a [i];
  }
  for (i=0; i < n; i++)
  { for (j=0; j < n-1; j++)
    { if (a[j] > a[j+1])
      { t = a[j];
        a[j] = a[j+1];
        a[j+1] = t;
      }
    }
  }
  cout << "Sorted array" << endl;
  for (i=0; i < n; i++)
  cout << a [i] << endl;
  getch();
}

```

Q. WAP to input elements in 2D array of size 3x3 & display them in matrix form.

```

#include <iostream.h>
#include <conio.h>
#include <iomanip.h>
void main ( )
{ int a [3][3], i, j;
  for (i=0; i < 3; i++)
  for (j=0; j < 3; j++)
  { cout << "Enter any no. ";

```

```

}
cin >> a[i][j];
}
}
char c;
cout << "Matrix : " << endl;
for (i=0; i<3; ++i)
{
for (j=0; j<3; ++j)
cout << setw(5) << a[i][j];
cout << endl;
}
}
}

```

Q. W.A.P to input elements in 2-D array & find out the sum of elements whose one's value is 2.

```

#include <iostream.h>
#include <conio.h>
#include <iomanip.h>
void main()
{
int i, j, a[4][4], s=0;
for (i=0; i<4; ++i)
for (j=0; j<4; ++j)
{
cout << "Enter any no. ";
cin >> a[i][j];
}
}
char c;
cout << "Sum = " << s;
}
}

```

Q. W.A.P to input 2-D array (4x4) & find out sum of both diagonals.

```

void main()
{
int a[4][4], i, j, s1=0, s2=0;
for (i=0; i<4; ++i)
for (j=0; j<4; ++j)
{
cout << "Enter any";
cin >> a[i][j];
}
}
for (i=0; i<4; ++i)
for (j=0; j<4; ++j)
{
if (i==j)
s = s + a[i][j];
}
}

```

20	21	22	23
20	11	12	13
20	20	22	23
20	21	22	23

Q. W.A.P to find sum of boundary elements in array n x n.

```

void main()
{
cout << "sum of d1 = " << s1;
cout << "sum of d2 = " << s2;
}
}
}
}

```

Logic

```

for (i=0; i<n; ++i)
{
for (j=0; j<n; ++j)
{
if (i==0 || j==0 || i==n-1 || j==n-1)
s = s + a[i][j];
}
}
}
}

```



Q. WAP to input 2D array of 4x4 & display the upper half matrix. (upper half is)

00	01	02	03
11	12	13	
	22	23	
		31	

```
#include <iostream.h>
```

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

```
#include <iomanip.h>
```

```
void upperhalf (int a [4][4])
```

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

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

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

```
if (j >= i)
```

```
cout << setw(5) << a[i][j];
```

```
else
```

```
cout << setw(5) << ' ';
```

```
cout << endl;
```

```
void main ()
```

```
{ int a[4][4], i, j;
```

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

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

```
{ cout << "Enter ";
```

```
cin >> a[i][j];
```

```
}
```

```
upperhalf (a);
```

```
getch();
```

Q. WAP to input name, Roll no. & Percentage of 10 students. Count & display the records of those student -s. who secures 80% or above using structures.

```
#include <iostream.h>
```

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

```
struct student;
```

```
{ char n[20];
```

```
int r;
```

```
float p;
```

```
};
```

```
void main ()
```

```
{ student s[10];
```

```
int c=0; int i;
```

```
for (i=0; i<10; ++i)
```

```
{ cout << "Enter the name";
```

```
cin >> s[i].n;
```

```
cout << "Enter Roll no";
```

```
cin >> s[i].r;
```

```
cout << "Enter Percentage";
```

```
cin >> s[i].p;
```

```
}
```

```
for (i=0; i<10; ++i)
```

```
if (s[i].p >= 80)
```

```
{ cout << s[i].n << s[i].r << s[i].p;
```

```
c = c++; }
```

```
cout << "Total No. of students  
above 80%" << c;
```

```
getch ();
```

```
}
```



# CHAPTER-7. DATA FILE HANDLING.

The main purpose of using file is to store our data in secondary storage namely of computer. Normally we store our data in RAM which is temporary storage. So whenever we store in RAM will wash out as we switch off the computer. So to store data permanently in secondary storage device, the file handling program is used.

Before working or writing the record from the file, the very first step is to open the file. There are two ways classes available to open the file.

1. ofstream
2. ifstream
3. fstream

These classes are declared within `<fstream.h>` header file. So, in every file handling program, `<fstream.h>` header file must be included. This header file itself includes `<ostream.h>` header file. So, there is no need to include `<ostream.h>` in file handling program.

ofstream  
To open the new file in output (writing) mode, `ofstream` class is used. If file is already existing then the existing records of that file will be erased & new one will be created.

## Syntax:-> ofstream filename ("external file name");

filever is an internal file name & can be any user define word having the surname of the buffer allocated for the program's count. file, so, data is stored in this buffer & external file name is the detail file name created on the disk to store the records of the file.

Ex -> ofstream f1 ("abc.txt");

if we want to store students record by creating a data file student.dat

```
#include <fstream.h>
#include <conio.h>
void main ()
{
  char n[20];
  int o;
  first p;
  ofstream f ("student.dat");
  char ch;
  do {
    cout << "Enter name, rollno, & percentage";
    cin >> n >> p;
    f << n << p;
    cout << "\n want to enter more records";
    cin >> ch;
  } while (ch == 'y');
  f.close ();
  getch ();
}
```



SYSTEM  
To open the existing data files to reading the records ifsystem class is used.

Syntax:- ifsystem filename ("external file name");

Q MAP to read student's record's from the file student.dat.

```
#include <fstream.h>
void main ()
{ char n [100];
  int p;
  float p;
  ifsystem f ("student.dat");
  while (f.eof () == 0)
  { f.get (>> >> p;
    cout << "Name = " << n << endl;
    cout << "Roll no. = " << r << endl;
    cout << "Percentage = " << p << endl;
  }
}
} getch ();
```

(eof () is end of file function)

Q. Write a function to read a content from a text file abc.txt, count & display the no. of capital letters in it.

```
void count ()
{ ifsystem f ("abc.txt");
  int c = 0;
```

```
char ch;
while (f.eof () == 0)
{ f.get (>> ch);
  if (ch >= 'A' && ch <= 'Z') or if (isupper (ch))
  c++;
}
```

```
f.close ();
cout << "No. of Capital letters = " << c;
getch ();
```

Q. MAP to read the content from a text file xyz.txt count & display no. of digits, no. of capital letter, no of small character, special character.

```
void count ()
{ ifsystem f ("xyz.txt");
  int d = 0, c = 0, s = 0, sp = 0;
  char ch;
  while (f)
  { f.get (>> ch);
    if (ch >= '0' && ch <= '9')
      d++;
    else if (ch >= 'A' && ch <= 'Z')
      c++;
    else if (ch >= 'a' && ch <= 'z')
      s++;
    else
      sp++;
  }
```

```
f.close ();
getch ();
```



Q. Write a function to read a content from the file lower.txt & write another file upper.txt after converting each small letter into capital letter.

```

void convert ()
{
    ifstream f1 ("lower.txt");
    ofstream o2 ("Upper.txt");
    char ch;
    while (f1.get (ch))
    {
        if (ch >= 'a' && ch <= 'z')
            ch = ch - 32;
        o2.put (ch);
    }
}
    
```

Q. WAF to count & display no. of lines stored with character A from a U file lower.txt.

```

void count ()
{
    ifstream f1 ("lower.txt"); int c=0;
    char ch;
    while (f1.get (ch))
    {
        if (ch == 'A')
            c++;
    }
    f1.close ();
}
    
```

Q. Count << No. of line stored with A << c; } getch ();

Q. WAF to read a content from a file story.txt Count & display the occurrence of 's' in the file story.txt;

```

void convert ()
{
    ifstream f1 ("story.txt");
    int i=0;
    char ch;
    while (f1.get (ch))
    {
        if (ch == 's')
            i++;
    }
    cout << i;
}
    
```

```

f1.close ();
cout << "No. of 's' is " << i;
getch ();
}
    
```



Q. Write to read the content from a file para.txt. Copy the content into the file para2.txt after replacing each 'M' into 'S'.

```
void convert()
{
    ifstream f1("para.txt");
    ofstream f2("para2.txt");
    char ch;
    while (f1.get(ch))
    {
        if (ch == 'M')
            ch = 'S';
        f2.put(ch);
    }
    f1.close();
    f2.close();
}

// Alternative solution
void copy()
{
    ifstream f1("note.txt");
    int c = 1;
    char ch;
    while (f1.get(ch))
    {
        if (ch == 'M')
            c--;
    }
}
```

else  
c = 1; if (c == 0)  
if (c == 0)  
f2.put(ch);  
f2.close();  
f1.close();  
getch();

fstream class by fstream class, we just declare the file variable.  
Syntax: fstream filename;

Open function. To open the file in different modes, open function is used.

Syntax: fstream.open("External file name", ios::mode);

When ios stands for input output streams. Modes: ios::in, ios::out, ios::app, ios::ate, ios::trunc, ios::plus, ios::binary.

(1) ios::out: To open a new file for writing the contents.

(2) ios::in: To open existing data file for reading the contents.



(3) ios::app To open the existing data file for appending the records.

(4) ios::trunc To open the existing data file & delete all the records.

(5) ios::ate It is used to open the existing data file & set the file pointer at last record.

(6) ios::beg It is used to open the file pointer at the beginning of the records.

(7) ios::binary It is used to open the file in binary form.

(8) ios::noexcept This causes to open function fail if the file does not already exist. It will not create a new file with that name.

Note Now when mode can be used with single open function.

e.g. -> ofstream f; f.open("student.dat", ios::out | ios::binary);

Close Function  
After completing input or output operation in the file. Finally the file will be close by using close();

Syntax: fileobj.close();

Input Output Operation

(i) Write ( )  
To transfer the content of record from bytes to disk data file, write ( ) is used.

Syntax: fileobj.write (char \* & record, size\_t (record));

where record is group variable which can be structure variable or class object. & size\_t will contain the size of variable in terms of memory bytes.

e.g. -> struct student {  
int x;  
float p;  
int y;  
};  
cout << size\_t (s);

When char \* is the cast operator



Q. WAP to create a datfile emp.dat which show name, code & basic pay of an employe.

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

struct emp
{
    char n[20];
    int code;
    float basic pay;
};

void main ( )
{
    emp e;
    fstream f;
    f.open ("emp.dat", ios::out | ios::binary);
    do {
        cout << "Enter name, code & basic pay";
        cin >> e.n >> e.c >> e.bp;
        f.write ((char *) &e, sizeof (e));
        cout << "\n want to enter more records";
        cin >> ch;
        while (ch == 'y');
        f.close ();
    }
    getch ();
}
```

(2) Read ( ) To read the record from existing data file & transfer it into buffer for processing, read () is used

SYNTAX:- fread (char \* &read, size\_t (records));

(3) eof ( ) (end of file)  
No check whether the file pointer is at end of the marker or not, eof () is used. This function returns 1 (true), if end of file mark encountered otherwise 0 (false).

SYNTAX:- fwrite (eof ( ) );

Q. WAP to send the content or records of employe from the file emp.dat.

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

struct emp
{
    char n[20];
    int c;
    float bp;
};

void main ( )
{
    emp e;
    fstream f;
    f.open ("emp.dat", ios::in, ios::binary);
    f.read ((char *) &e, sizeof (e));
    while (f.eof() == 0)
    {
        cout << "Name = " << e.n;
        cout << "Code = " << e.c;
        cout << "Basic pay = " << e.bp;
    }
}
```







```

} e.output (c);
f.stdout (char*) &e, sizeof(e));
}
}
f.close (c);
break;
}
case 's':
cin >> m;
f.open ("emp.dot" ios::in | ios::binary);
cout << "Enter code to be searched ";
while (f.eof() != 0)
{
if (m == e.returncode (c))
e.output (c);
f.stdout (char*) &e, sizeof(e));
}
f.close (c);
break;
}

```

```

case 'm':
case 'N':
f.open ("emp.dot" ios::in | ios::out | ios::binary);
cout << "Enter code to be modified";
cin >> m;
f.stdout (char*) &e, sizeof(e));
int c = 0;
while (f.eof() != 0)
{
c++;
if (m == e.returncode (c))
{
e.input (c);
f.seekp ((c-1) * sizeof(e));
f.write (char*) &e, sizeof(e));
}
f.stdout (char*) &e, sizeof(e));
}
}
f.close (c);
break;
case 'd':
case 'D':
fstream t;
f.open ("emp.dot" ios::in | ios::binary);
t.open ("temp.dot" ios::out | ios::binary);
cout << "Enter code to be deleted";
cin >> m;
f.stdout (char*) &e, sizeof(e));
while (f.eof() != 0)
{
if (t.write (char*) &e, sizeof(e));
f.stdout (char*) &e, sizeof(e));
}
}

```







for writing  
for reading  
seek ( ) & seekg ( )

seek ( ) & seekg ( ) are used to set the file pointer at particular byte no.

file no. seekpos seek ( n );

where n is the byte no. where we want to set the file pointer

seek ( ) can be used in two ways :-

(i) by absolute position

e.g. :- f.seek ( 30 );

It will move the get file pointer to byte no. 30.

(ii) By relative position

It can be used by 3 ways :-

(a) ios :: beg  
e.g. :- f.seek ( 20, ios :: beg );

It will move the file pointer from the 20 byte no. suffix to the beginning from the file.

beginning to 20 byte no.

(b) ios :: cur

It suffix from the current position from the file.

e.g. :- f.seek ( 10, ios :: cur );

It will move the file pointer, 10 bytes from the current location.

e.g. :- f.seek ( -10, ios :: cur );

(c) ios :: end

It suffix to the end of the file.

e.g. :- f.seek ( -5, ios :: end );

Q. W.A.F to count the no. of words present in a text file named : exam.txt. Assume that each word is separated by a single space & no blank space in the beginning & end of.