

## String Function

String function is a library function which is used to find the length of given string, reverse the string, copy the string, change into upper case or lower case, compare the two strings, etc.

There are 7 string function. They are:

1. strlen()
2. strrev()
3. strcat()
4. strupr()
5. strlwr()
6. strcmp()
7. strcpy()

### 1. strlen()

This string function is used to find the length of string. Its keyword is `strlen()`.

For example:

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

```

int x ;
char word[50];
printf("enter a word ");
scanf("%s", word);
x = strlen(word);
printf("The length of word is %d ", x);
getch();

```

{3}

**Output**

Enter a word prapti  
The length of word is 6

**2. Strrev()**

This string function is used to reverse the given string.

For example:

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
Void main()
{
    clrscr();
    char word[25];
    printf("enter a word\n");
    scanf("%s", word);
    strrev(word);
    printf("The reversed word is %s", word);
}

```

```
getch();
}
```

**Output**

Enter a word

prapti

The reversed word is itparp

**3. strcat()**

This string function is used to join two string together.

For example:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    clrscr();
    char word1[25], word2[25];
    printf("enter two words\n");
    scanf("%s %s", word1, word2);
    strcat(word1, word2);
    printf("The combined word is %s", word1);
    getch();
}
```

**Output**

Enter two words

gita , sita

The combined word is gitasita

#### 4. strupr()

The string function is used to convert lower case into upper case letter.

For example:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    clrscr();
    char word[25];
    printf("enter a word\n");
    scanf("%s", word);
   strupr(word);
    printf("The upper case letter is %s",
           word);
    getch();
}
```

**Output**

Enter a word

prapti

The upper case letter is PRAPTI

## S. Strlwr()

The string function is used to convert upper case into lower case letter.

For example:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    clrscr();
    char word[25];
    printf("enter a word\n");
    scanf("%s", word);
    strlwr(word);
    printf("The lower case letter is
           %s", word);
    getch();
}
```

### Output

Enter a word

PRAPTI

The lower case letter is  
prapti

## 6. strcmp()

This string function is used to compare two strings.

For example:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    clrscr();
    char word1[25], word2[25];
    printf("enter two words\n");
    scanf("%s %s", word1, word2);
    if(strcmp(word1, word2) == 0)
    {
        printf("Both string are same");
    }
    else
    {
        printf("Both string are not same");
    }
    getch();
}
```

**Output**  
 Enter two words  
 prapti, prapti  
 Both string are same

## 7. strcpy()

This string function is used to copy string from one string variable to another variable.

For example:

```
#include<stdio.h>
#include<conio.h>
#include <string.h>
void main()
{
    clrscr();
    char word1[25], word2[25];
    printf("enter a word\n");
    scanf("%s", word2);
    strcpy(word1, word2);
    printf("The copied word is %s", word1);
    getch();
}
```

### Output

enter a word

prapti

The copied word is prapti

Q. WAP to check whether the given word is palindrome or not.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    clrscr();
    Char word1[25], word2[25];
    printf("enter a word\n");
    scanf(".\n.s", word1);
    strcpy(word2, word1);
    strrev(word1);
    if(strcmp(word1, word2)==0)
    {
        printf("palindrome");
    }
    else
    {
        printf("Not palindrome");
    }
    getch();
}
```

### Output

Enter a word  
prapti  
NOT palindrome

## Array

An array is a special variable that can store more than one similar type of value under the same name with different index numbers.

### Syntax :

datatype arrayName[array-size];

→ int a[10]

- Q. Write a C program to ask 5 numbers & display the same numbers.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int a[5], i;
    for(i=0, i<5; i++)
    {
        printf("enter %.d numbers\n", i+1);
        scanf("%d", &a[i]);
    }
    printf("The numbers you have entered
           are \n");
}
```

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

### Output

Enter 1 number

5

Enter 2 number

3

Enter 3 number

4

Enter 4 number

2

Enter 5 number

1

The number you have  
entered are 5 3 4 2 1

of 10 students

~~H.W Q.~~ WAP to ask computer marks & display the mark which is increased by 2.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int mark[10], i;
    for (i=0; i<10; i++)
    {
        printf("enter mark of %d student\n",
               i+1);
        scanf("%d", &mark[i]);
    }
    printf("The marks after adding 2
           marks on previous mark are\n");
    for (i=0; i<10; i++)
    {
        printf("%d", mark[i]+2);
    }
    getch();
}
```

## Output

Enter the mark of 1 student

33

Enter mark of 2 student

38

enter mark of 3 student

43

enter mark of 4 student

48

enter mark of 5 student

53

enter mark of 6 student

42

enter mark of 7 student

29

enter mark of 8 student

10

enter mark of 9 student

98

enter mark of 10 student

65

The marks after adding 2 marks on  
previous mark are 35 40 45 50 55  
44 31 12 100 67

- Q. Write a C program to ask 2 matrix of order  $2 \times 2$  and display the matrix addition.

```
#include <stdio.h>
#include <conio.h>
void main()
{
    clrscr();
    int A[2][2], B[2][2], C[2][2],
        i, j;
    printf("enter matrix A\n");
    for(i=0; i<2; i++)
    {
        for(j=0; j<2; j++)
        {
            scanf("%d", &A[i][j]);
        }
    }
    printf("enter matrix B\n");
    for(i=0; i<2; i++)
    {
        for(j=0; j<2; j++)
        {
            scanf("%d", &B[i][j]);
        }
    }
    for(i=0; i<2; i++)
    {
        for(j=0; j<2; j++)
        {
            C[i][j] = A[i][j] + B[i][j];
        }
    }
    printf("addition of two matrices is\n");
    for(i=0; i<2; i++)
    {
        for(j=0; j<2; j++)
        {
            printf("%d ", C[i][j]);
        }
        printf("\n");
    }
}
```

```

for (j=0; j<2; j++)
{
    c[i][j] = A[i][j] + B[i][j];
}
printf("The addition is\n");
for (i=0; i<2; i++)
{
    for(j=0; j<2; j++)
    {
        scanf ("%d", &c[i][j]);
        printf("\n");
    }
    getch();
}

```

Output  
enter matrix A

1  
2  
3  
4

enter matrix B

5  
6  
7  
8

The addition is  
6 8  
10 12

Q. Write a C program to ask 2. matrix of order  $3 \times 3$  and display the matrix subtraction.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int A[3][3], B[3][3], C[3][3], i, j;
    printf("enter matrix A\n");
    for (i=0; i<3; i++)
    {
        for (j=0; j<3; j++)
            scanf("%d", &A[i][j]);
    }
    printf("enter matrix B\n");
    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] = A[i][j] - B[i][j];
    }
    printf("Resultant matrix C\n");
    for (i=0; i<3; i++)
    {
        for (j=0; j<3; j++)
            printf("%d ", C[i][j]);
        printf("\n");
    }
}
```

```

{  

    C[i][j] = A[i][j] - B[i][j];  

}

```

y

y

```
printf("The subtraction is\n");
```

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

{

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

{

```
        scanf("%d", &C[i][j]);  
        printf("\n");
```

}

getch();

y

### Output

enter matrix A

1  
2  
3  
4  
5  
6  
7  
8  
9

enter matrix B

1  
2  
3  
4  
5  
6  
7  
8  
9

The subtraction is

0	0	0
0	0	0
0	0	0

Q. WAP to ask salaries of 10 employee & count the number of salaries between 15000 & 20,000.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int salary[10], i, count = 0 ;
    for (i=0; i<10; i++)
    {
        printf("enter salary of %d employee\n", i+1);
        scanf("%d", &salary[i]);
        if(salary[i] > 15000 && salary[i] < 20000)
            count++;
    }
    printf("The number of salaries between
           15000 & 20000 is %d", count);
    getch();
}
```

Output to printing was as follows

enter salary of 1 employee  
6000

enter salary of 2 employee  
14000

enter salary of 3 employee  
18000

-----  
-----  
-----  
-----  
-----

enter salary of 10 employee  
6000

The number of salaries between is 0000 &  
20000 is 3.

- Q. WAP to ask salary of 20 employee & find the largest salary among them.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int salary[20], i;
    printf("enter the salary\n");
    for(i=0; i<20; i++)
    {
        scanf("%d", &salary[i]);
    }
    int largest_salary = 0;
    for(i=0; i<20; i++)
    {
        if(salary[i] > largest_salary)
        {
            largest_salary = salary[i];
        }
    }
    printf("the maximum salary is RS. %d", largest_salary);
    getch();
}
```

Output

Enter the salary:

20,000

30,000

-

-

-

The largest salary is RS.

30,000

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

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

```
void main()
```

```
{
```

```
clrscr();
```

```
int salary[20], i, largest;
```

```
for(i=0; i<20; i++)
```

```
{
```

```
printf("enter salary of %d employee\n", i+1);
```

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

```
}
```

```
largest = salary[0];
```

```
for(i=0; i<20; i++)
```

```
{
```

```
if(largest<salary[i])
```

```
{
```

```
largest = salary[i];
```

```
y  
y  
printf("The largest salary is %.d ", largest);  
getch();  
y
```

output

Enter salary of 1 employee

6000

Enter salary of 2 employee

25000

-

-

-

Enter salary of 20 employee

16000

The largest salary is

25000

c.u2 \* WAP to ask computer mark of 10 students  
and display the lowest mark among them.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int salary[10], i, lowest;
    for(i=0; i<10; i++)
    {
        printf("enter salary of %d student", i+1);
        scanf("%d", &salary[i]);
    }
    lowest = salary[0];
    for(i=0; i<10; i++)
    {
        if(lowest>salary[i])
        {
            lowest = salary[i];
        }
    }
    printf("The lowest mark is %d", lowest);
    getch();
}
```

Output

enter mark of 1 student

5000

enter mark of 2 student

42

enter mark of 10 student

3

The lowest mark is 3

VVI

Q. WAP to ask 10 numbers & display all numbers in ascending order.

```
#include<stdio.h>
#include<conio.h>
Void main()
{
    clrscr();
    int n[10], i, j, temp;
    for(i=0; i<10; i++)
    {
        printf("enter .d number\n", i+1);
        Scanf(".d", &n[i]);
    }
}
```

```
for(i=0 ; i<10 ; i++)  
{  
    for(j=i+1 ; j<10 ; j++)  
    {  
        if(n[i]>n[j])  
        {  
            temp=n[i];  
            n[i]=n[j];  
            n[j]=temp;  
        }  
    }  
}  
printf("The numbers after arranging in  
ascending order are\n");  
for(i=0 ; i<10 ; i++)  
{  
    printf("%d", n[i]);  
}  
getch();  
}
```

Output

enter 1 number

10

enter 2 number

3

=

Enter 10 number

12

The numbers after arranging in ascending order are

3 - - - 10 12

Q. WAP to ask salary of 10 employees and display all the salaries from highest to lowest. ( highest salaries to lowest salaries)

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int salary[10], i, j, temp;
    for(i=0; i<10; i++)
    {
        printf("enter .d salary\n", i+1);
        scanf(".d ", &n[i]);
    }
    for(i=0; i<10; i++)
    {
        for(j=i+1; j<10; j++)
        {
            if(n[i] > n[j])
            {
                temp = n[i];
                n[i] = n[j];
                n[j] = temp;
            }
        }
    }
}
```

```
n[i]=n[j];
n[j]=temp;
}
}
printf("The numbers of salaries from
highest to lowest are\n");
for(i=0;i<10;i++)
{
printf("%d ",n[i]);
}
getch();
```

Output

enter 1 salary

5000

enter 2 salary

8000

-

-

enter 10 salary

2000

The numbers of salaries from  
highest to lowest are

8000, 5000, ..., 2000

Q. WAP to ask computer mark of 20 students & count the number of pass and failed students.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    int mark[20], i, pass=0, fail=0;
    for(i=0; i<20; i++)
    {
        printf("enter mark of %d student ", i+1);
        scanf("%d", &mark[i]);
        if(mark[i]>=40)
        {
            pass++;
        }
        else
        {
            fail++;
        }
    }
    printf("pass students=%d\n", pass);
    printf("fail students=%d\n", fail);
    getch();
}
```

Output

enter mark of 1 student

50

enter mark of 2 student

20

-

-

-

enter mark of 20 student

18

pass students = 18

fail students = 2

## Chapter 4

# Function

### Concept of function

A function is a self-contained program segment or the block of statements that perform some specific, well-defined task. Every C program ~~contains~~ comprises one or more such functions. The execution of any program always begins by executing the instructions in the main () functions.

### Types of Function

#### a. Library / Built-in Functions

#### b. user-defined functions

#### a. Library / Built-in functions

The library function is defined by the developer of C-language. It cannot be changed or altered. We can use library function by simply adding a header file.

For example:

1. printf(), scanf() are defined in header file stdio.h

2. getch(), clrscr() are defined in header file conio.h

3. strlen(),strupr() are defined in header file string.h

4. pow(),sqrt() are defined in header file math.h

### b. User-defined Function

This type of function is defined by the user or programmer as per their need. We can use a user-defined function by simply declaring a function.

For example: `aor()`, `vol()`, `sum()`, `factorial()`, etc.

#### \* Categories of user-defined function

1. Functions without any arguments and return values
2. Function with arguments but no return values
3. Function with arguments and with return values
4. Function without any arguments but with return values

#### 1. Function without any arguments and return values

Here, we simply call a function without passing arguments and the called function doesn't have to return any values to the calling function. If we do not want to return a value, we must use the return type `void` and miss out on the return statement.

For example:

`//WAP to find the area of rectangle using  
function.`

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

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

```
void AOR();
```

```
void main()
```

```
{  
clrscr();  
AOR();  
getch();  
}  
  
void AOR()  
{  
int l,b,area;  
printf (" write the length\n");  
scanf ("%d ", &l);  
printf (" write the breadth\n");  
scanf ("%d ", &b);  
area=l*b;  
printf ("The area of rectangle is %d ,area);  
}  
}
```

### Output

en: write the length

5

write the breadth

10.

The area of rectangle is 50

with return value = int  
without return value = void  
with arguments {---}  
without arguments {}

Date : .....

Page : .....

## Homework

// WAP to find the area of four walls. area = 2 h(l+b).

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

Output

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

enter length

```
void AOFW();
```

2

```
void main()
```

enter breadth

```
{
```

3

```
clrscr();
```

enter height

```
AOFW();
```

4

```
getch();
```

The area of four wall

```
3
```

is 40

```
void AOFW()
```

```
{
```

```
int l,b,h,Area of four wall;
```

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

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

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

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

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

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

Area of four wall =  $2 * h * (l + b)$

```
printf("The area of four wall is %d", area of four  
wall)
```

```
3
```

2. Function with arguments but no return values  
This type of function has one-way communication.  
Here, an argument is passed from the calling function to the called function, but there is no need for a return statement in the called function.

For example:

```
// WAP to find area of rectangle using function
#include<stdio.h>
#include<conio.h>
void NOR(int,int);
void Main()
{
    clrscr();
    int l,b;
    printf("enter length & breadth\n");
    scanf("%d %d", &l, &b);
    NOR(l,b);
    getch();
}
void NOR(int l, int b)
{
    int area;
    area = l*b;
    printf("The area of rectangle is %d", area);
}
```

3. Function with arguments and with return values
- This type of function has two-way communication. Here, an argument is passed from the calling function to the called function, and there will also be returning statement in the called function.

For example:

```
#include <stdio.h>
#include <conio.h>
int AOR(int, int);
void main()
{
    clrscr();
    int l, b, x;
    printf("enter length and breadth\n");
    scanf("%d %d", &l, &b);
    x = AOR(l, b);
    printf("The area of rectangle is %d ", x);
    getch();
}

int AOR(int l, int b)
{
    int area;
    area = l * b;
    return area;
}
```

4. Function without any arguments but with return values  
This type of function also has one-way communication.  
Here, an argument is not passed from the calling function to the called function, but there is a need for a return statement in the called function.

For example :

```
#include <stdio.h>
#include <conio.h>
int AOR();
void main()
{
    clrscr();
    int x;
    x = AOR();
    printf("The area of rectangle is %.d", x);
    getch();
}

int AOR()
{
    int l,b,area;
    printf("enter length and breadth\n");
    scanf("%d %d",&l,&b);
    Area = l*b;
    return area;
}
```

**Homework from Exam question - 5 Marks****Question:**

V.V.J.  
last year  
# //WAP to ask a radius of a football and  
find the surface area of football. [area =  $4\pi r^2$ ]

```
#include<stdio.h>
#include<conio.h>
float AOF(float);
void main()
{
    clrscr();
    float r, x ;
    printf("enter a radius\n");
    scanf("r.f ", &r);
    x = AOF(r);
    printf("The surface area of
    football is %f", x);
    getch();
}
```

```
float AOF(float r)
{
    float area , pi ;
    pi = (float) 22 / 7 ;
    area = 4 * pi * r * r ;
    return area;
```

**Output**

Enter a radius

7

The surface area of
football is 616.00000

## # Components of function

1. Function prototype (Function declaration)
2. Function call
3. Function Definition
4. return and void statements of a function.

## # Actual argument and Formal argument

Arguments defined inside the function are called formal arguments. The argument from which the arguments have been passed to a function is known as actual arguments.

## # concept of recursion

A recursive function is one, which called itself.

A recursive function contains following features:

1. The function should call itself.
2. The function should have a stopping condition.

The concept of using the recursive function to repeat the execution of statement many times is known as recursion.

For example:

// WAP to find the factorial of a given number using recursive function .

VVV  
and  
term

```
#include <stdio.h>
#include <conio.h>
int factorial(int);
void main()
{
    clrscr();
    int n, fact;
    printf("enter a number\n");
    scanf("%d", &n);
    fact = factorial(n);
    printf("The factorial of %d is %d", n, fact);
    getch();
}

int factorial(int n)
{
    if (n == 0 || n == 1)
    {
        return 1;
    }
    else
    {
        return (n * factorial(n - 1));
    }
}
```

Output  
enter a number

2

The factorial of 2 is 2

QSN: Write a C program to calculate the sum of all natural numbers upto that num-a given number using recursive function.

```
#include<stdio.h>
#include<conio.h>
int sum(int);
void main()
{
    clrscr();
    int n, x;
    printf("enter a number\n");
    scanf("%d", &n);
    x = sum(n);
    printf("The factorial sum is %d", x);
    getch();
}

int sum(int n)
{
    if (n == 0)
        {
            return 0;
        }
    else
        {
            return (n + sum(n - 1));
        }
}
```

## 1. Structure

A structure is the collection of one or more variable of different datatypes (e.g: int, float, char), grouped together under a single name. We can have an array of structure. The keyword struct is used to declare structure.

for example,

```
/* WAp to store five employees records (EID, name,  
post and department) and display them using  
structure . */
```

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

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

```
Struct employee
```

```
{
```

```
int eid;
```

```
char name[50];
```

```
char post[50];
```

```
char dep[50];
```

```
} emp[5];
```

```
void main()
```

```
{
```

```
clrscr();
```

```
int i;
```

```
for(i=0;i<5;i++)
```

```
{
```

```
printf("Enter eid, name, post and department of 4rd  
employee\n"), i+1);  
scanf("%d %s %s %s", &emp[i].eid, emp[i].name,  
emp[i].post, emp[i].dep);  
}  
clrscr();  
printf("The records of employees are listed below\n");  
printf("EID\tName\tPost\tDepartment\n");  
for(i=0; i<5; i++)  
{  
    printf(" %d\t%s\t%s\t%s\n", emp[i].eid,  
          emp[i].name, emp[i].post, emp[i].dep);  
}  
getch();  
}
```

```

/* WAP to store 3 students records (grade, rollno,
name and faculty) and display them using structures */
#include <stdio.h>
#include <conio.h>
struct students
{
    int grade;
    int rollno;
    char name[50];
    char faculty[50];
} std[3];
void main()
{
    clrscr();
    int i;
    for(i=0; i<3; i++)
    {
        printf("Enter grade, rollno, name and faculty of %d
student\n", i+1);
        scanf("%d %d %s %s", &std[i].rollno, &std[i].grade, &std[i].
name, std[i].faculty);
    }
    clrscr();
    printf("The records of students are listed below\n");
    printf("Grade\t\tRollno\t\tName\t\tFaculty\n");
    for(i=0; i<3; i++)

```

```

{
printf ("%d\t%d\t%d\t%s\n", std[i].  

    grade, std[i].rollno, std[i].name, std[i].  

    faculty);
}
getch();
}

```

Q1 /\* WAp to store five employees records( EID, name, post and salary) and display all the records whose salary is greater than 5000 using structure.\* /

```

#include<stdio.h>
#include<conio.h>
struct employee
{
    int eid ;
    char name[50];
    char post[50];
    int sal;
} emp[5];
void main()
{
    clrscr();
    int i;
    for(i=0;i<5;i++)

```

{

printf("Enter eid, name, post and salary of %d  
employee\n", i+1);

scanf("%d %s %s %d", & emp[i].eid, emp[i].name,  
emp[i].post, & emp[i].sal);

}

clrscr();

printf("The records of employees whose salary >  
5000 are listed below\n");

printf("EID\tName\tPost\tSalary\n");

for (i=0; i&lt;5; i++)

{

if (emp[i].sal &gt; 5000)

{

        printf("%d\t%s\t%s\t%d\n", emp[i].eid,  
                emp[i].name, emp[i].post, emp[i].sal);

}

getch();

}

IMP

\* WAP to store five students records (roll no, name and marks in 3 subjects) and display all the students who passed in exam using structure.\*/

```
#include<stdio.h>
#include<conio.h>
struct student
{
    int roll_no;
    char name[50];
    int eng, nep, com;
} std[5];
void main()
{
    clrscr();
    int i;
    for(i=0; i<5; i++)
    {
        printf("enter roll no, name , marks in eng, nep
and com of %d students\n", i+1);
        scanf("%d %s %d %d %d", &std[i].roll_no, std[i].
name, &std[i].eng, &std[i].nep, &std[i].
com);
    }
    clrscr();
    printf("The records of students who passed in exam are
listed below\n");
}
```

```
printf("Rollno\t Name\t Eng\t Nep\t COM\n");
for(i=0;i<5;i++)
{
    if(std[i].eng>=40 && std[i].nep>=40 && std[i].com>=40)
    {
        printf("%d\t %s\t %d\t %d\t %d\n", std[i].rollno, std[i].name, std[i].eng, std[i].nep,
               std[i].com);
    }
}
getch();
}
```

Last year question:

/\* WAP to store five employees records (eid, name, post and salary) and display all the records whose salary is between 25000 and 40000.

```
#include <stdio.h>
#include <conio.h>
Struct employee
{
    int eid;
    #char name[50];
    char post[50];
    int sal;
}emp[5];
void main()
{
    clrscr();
    int i;
    for (i=0; i<5; i++)
    {
        printf("Enter eid, name, post & and salary of %d employee\n", i+1);
        scanf("%d %s %s %d", &emp[i].eid, emp[i].name,
              emp[i].post, &emp[i].sal);
    }
    clrscr();
    printf("The records of employees whose salary is
           between 25000 and 40000 are\n");
}
```

```

printf("eid\tname\t post\t salary\n");
for(i=0; i<5; i++)
{
    if(emp[i].sal>25000 && emp[i].sal<40000)
    {
        printf("%d\t %s\t %s\t %d\n", emp[i].eid,
               emp[i].name, emp[i].post, emp[i].sal);
    }
}
getch();
}

```

~~V.V.2~~

/\*WAP to store five employees records( eid, name, post and salary) and display all the records from highest to lowest salary \*/

```

#include<stdio.h>
#include<conio.h>
#include <string.h>
struct employee
{
    int eid;
    char name[50];
    int sal;
}emp[5];

```

```
void main()
```

```
{
```

```
clrscr();
```

```
int i, j;
```

```
int tempeid, tempsal;
```

```
char tempname[50];
```

```
for (i=0; i<5; i++)
```

```
{
```

```
printf(" Enter eid, name and salary of %d employee in",  
      i+1);
```

```
scanf("%d %s %d", &emp[i].eid, emp[i].name,  
      &emp[i].sal);
```

```
}
```

```
clrscr();
```

```
for (i=0; i<5; i++)
```

```
{
```

```
for (j=i+1; j<5; j++)
```

```
{
```

```
if (emp[i].sal < emp[j].sal)
```

```
{
```

```
tempeid = emp[i].eid;
```

```
emp[i].eid = emp[j].eid;
```

```
emp[j].eid = tempeid;
```

```
strcpy(tempname, emp[i].name);
```

```
strcpy(emp[i].name, emp[j].name);
```

```
strcpy(emp[j].name,tempname);
```

```
tempsal=emp[i].sal;
```

```
emp[i].sal=emp[j].sal;
```

```
emp[j].sal=tempsal;
```

```
}  
}
```

```
printf("The records of employees from highest to  
lowest salary are\n");
```

```
printf("eid\tname\tsalary\n");
```

```
for(i=0;i<5;i++)
```

```
{
```

```
printf("%d\t%s\t%d\n",emp[i].eid,emp[i].  
name,emp[i].sal);
```

```
}
```

```
getch();
```

```
y
```

H.W

- \* WAP to ask rollno, name and computer mark of 10 students and display all the records from lowest to highest marks.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
struct student
{
    int rollno;
    char name[50];
    int comp;
} std[10];
void main()
{
    clrscr();
    int i, j;
    int temprollno, tempcomp;
    char tempname[50];
    for(i=0; i<10; i++)
    {
        printf("Enter rollno, name and computer mark of %d\n"
               "student", i+1);
        scanf("%d %s %d", &std[i].rollno, std[i].name, &
              std[i].comp);
    }
}
```

```

clrscr();
for(i=0; i<10; i++)
{
    for(j=i+1; j<10; j++)
    {
        if (std[i].comp > std[j].comp)
        {
            temprollno = std[i].rollno;
            std[i].rollno = emp[j].rollno;
            std[j].rollno = temprollno;

            strcpy(tempname, emp[i].name);
            strcpy(std[i].name, emp[j].name);
            strcpy(std[i].name, tempname);

            tempcomp = std[i].comp;
            std[i].comp = std[j].comp;
            std[j].comp = tempcomp;
        }
    }
}

printf("The records of students computer marks from highest to
lowest to highest marks are\n");
printf("%d\t %s\t %d\t %d\n", std[i].rollno, std[i].name, std[i].comp);
for(i=0; i<10; i++)
{
    printf("%d\t %s\t %d\t %d\n", std[i].rollno, std[i].name, std[i].comp);
}
getch();
}

```

# WAP to store five employees records (eid, name and salary) and display all the records in alphabetical order by name using structures\*/

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
struct employee
{
    int eid;
    char name[50];
    int sal;
}emp[5];
void main()
{
    clrscr();
    int i,j;
    int tempeid,tempal;
    char tempname[50];
    for(i=0;i<5;i++)
    {
        printf("enter eid , name and salary of %d employee\n",
               i+1);
        scanf("%d %s %d", &emp[i].eid, emp[i].name &
              emp[i].sal);
    }
}
```

```

clrscr();
for(i=0; i<5; i++)
{
    for(j=i+1; j<5; j++)
    {
        if(strcmp(emp[i].name, emp[j].name)>0)
        {
            tempeid = emp[i].eid;
            emp[i].eid = emp[j].eid;
            emp[j].eid = tempeid;
        }
    }
}

```

```

strcpy(tempname, emp[i].name);
strcpy(emp[i].name, emp[j].name);
strcpy(emp[j].name, tempname);

```

```

tempsal = emp[i].sal;
emp[i].sal = emp[j].sal;
emp[j].sal = tempsal;
}
}
}

```

```

printf("The records of employees in alphabetical
order are\n");
printf("eid\tname\ttsal\n");
for(i=0; i<5; i++)
{
}

```

```

printf("%d%d%d", emp[i].eid, emp[i].
      name, emp[i].sal);
}
getch();
}

```

*/\* WAP to read the students name and mark  
of 'n' students and display the students whose  
mark is above 50 and also count it. \*/*

```

#include<stdio.h>
#include<conio.h>
Struct student
{
    char name[50];
    int mark;
};
void main()
{
    clrscr();
    int i, count=0, n);
    printf("how many student records you want
           to enter\n");
    scanf("%d", &n);
    Struct student std[n];
    for(i=0; i<n; i++)

```

```
{  
printf("Enter name and mark of %d student\n",  
      i+1);  
scanf("%s %d", std[i].name, &std[i].mark);  
}  
(1)scanf();  
printf("The records of students whose marks is  
above 50 are\n");  
printf("Name\tMark\n");  
for(i=0; i<n; i++)  
{  
if(std[i].mark > 50)  
{  
printf("%s\t%d\n", std[i].name, std[i].mark);  
count++;  
}  
}  
printf("The total record is %d", count);  
getch();  
}
```

## File handling

### 1. Why do you need file?

→ As we know, while the program is processing the content of variable are stored in RAM but the problem is if such values of variable are needed in future, then it comes this causes the problem because the value of the variable is only within the program is being processed. After a program is over, the content of the RAM is erased. To overcome this problem we can use datafile to save the content into it, so that we can retrieve content from data file whenever we want it.

### 2. WAP to read name & salary of 10 employees and store all the data in a file named "emp.txt" and also display all the records from the file to the output screen.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i, sal;
    char name[50];
    FILE * fp;
```

~~fp = file~~

```
fp = fopen("emp.txt", "w");
for(i=0; i<10; i++)
{
    printf("Enter name & salary of %d employee\n", i+1);
    scanf("%s %d", name, &sal);
    printf(fp, "%s %d", name, sal);
}
```

```
fclose(fp);
```

```
printf("The records of employees are\n");
printf("Name\tSalary\n");
```

```
fp = fopen("emp.txt", "r");
for(i=0; i<10; i++)
{

```

```
fscanf(fp, "%s %d", name, &sal);
```

```
printf("%s\t%d\n", name, sal);
}
```

```
getch();
}
```

/\* WAP to read rollno, name and address of some students according to the need of user and display the records from the file to the output screen \*/

```
#include<stdio.h>
#include <conio.h>
#include <string.h>
void main()
{
    clrscr();
    int i, rollno;
    char name[50], address[50], choice[50];
    FILE *fp;
    fp = fopen("std.txt", "w");
    do
    {
        printf("Enter rollno, name and address of a student\n");
        scanf("%d %s %s", &rollno, name, address);
        fprintf(fp, "%d %s %s\n", rollno, name, address);
        printf("Do you want to record enter more data(y/N)\n");
        scanf("%s", choice);
        if(choice == 'y')
    } while(strcmp(choice, "y") == 0);
    fclose(fp);
    printf("The records of students are\n");
}
```

```

printf("Rollno\tName\tAddress\n");
fp=fopen("std.txt","r");
while(fscanf(fp,"%d\t%s\t%s", &rollno, name,
address)!=EOF)
{
    printf ("%d\t%s\t%s\n", rollno, name, address);
}
fclose(fp);
getch();
}

```

Output :

Enter rollno, name and address of  
a student

1

prapti

Ktm

Do you want to enter more data(y/N)

y

Enter rollno, name and address of a student

2

priya

dhawan

Do you want to enter more data(y/N)

n

The records of students are

Rollno	Name	Address
1	Prapti	Ktm
2	Priti	Dhawan

\* Write a program to create and write data into file.[4 marks] \*/

```
#include <stdio.h>
#include <stroconio.h>
#include <string.h>
void main()
{
    clrscr();
    int i, rollno;
    char name[50], address[50], choice[50];
    FILE *fp;
    fp=fopen("std.txt","w");
    do
    {
        printf("Enter rollno, name and address of a
               student\n");
        scanf("%d %s %s", &rollno, name, address);
        fprintf(fp, "%d %s %s\n", rollno, name,
                address);
    } while(choice != 'n');
}
```

```

printf("Do you want to enter more data(y/n)\n");
scanf("\%s", choice);
strupr(choice);
if(strcmp(choice, "y") == 0) {
    fclose(fp);
    printf("All data are saved successfully");
    getch();
}

```

*/\* WAP to read data from the file and display it in the output screen using file handling functions \*/*

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    clrscr();
    int i, Rollno;
    char name[50], address[50], choice[50];
    FILE *fp;
    printf("The records of students are\n");
    printf("Rollno\tName\tAddress\n");
    fp = fopen("std.txt", "r");
    while (scanf(fp, "%d\t%s\t%s", &Rollno, name,
        address) != EOF)

```

{

printf("%d\t%s\t%d\n", rollno, name, address);

}

fclose(fp);

getch();

}

/\* WAP to read id, name and salary of some employee according to the need of user and display the records whose salary is between 10000 and 15000 from the file to the output screen.

#include &lt;stdio.h&gt;

#include &lt;conio.h&gt;

void main()

{

clrscr();

int i, id, salary;

char name[50], choice[50];

FILE \*fp;

fp=fopen("emp.txt","w");

do

{

printf("Enter id, name and salary of employee\n");

scanf("%d %s %d", &amp;id, name, &amp;salary);

fprintf(fp,"%d\t%s\t%d\n", id, name, salary);

printf("Do you want to enter more data (Y/N)\n");

```
scanf("y.s", choice);
strupr(choice);
} while(strcmp(choice, "y") == 0);
fclose(fp);
printf("The records of employees are\n");
printf("Edit name & salary\n");
fp = fopen("emp.emp.txt", "r");
while(fscanf(fp, "%d %s %d", &id, name, &
salary) != EOF)
{
    if(salary > 10000 && salary < 15000)
        printf("%d %s %d\n", id, name, salary);
}
fclose(fp);
getch();
}
```

Q. Describe file handling operations in C? [4 mark]

→ These are different ways in which file is to be opened:

The different file opening modes are:

1. r

→ In this mode, the file is opened for reading the existing file.

2. w

→ In this mode, the file is opened for writing. If the file exists, previous contents will be erased and if the file doesn't exist, previous contents will be erased and if the file doesn't exist, then file will be created.

3. a

→ In this mode, the file is opened to append if the file exists but if the file doesn't exist, the new file will be created.

4. r+

→ In this mode, the file is opened for both reading and writing. The file must already exist before performing the operation.

5.  $w^+$

→ In this mode, the file is opened for both writing and reading.

6.  $at$

→ In this mode, the file is opened for both reading and appending.

Q. what is pointer? why should we use a pointer?

→ A pointer is a variable that holds memory location of another variable rather than actual value. we use pointer variable because it stores a address or memory location and points to whatever that memory location contains.

we use a pointer for :- / Advantages:

1. pointers enable us to access a variable that is defined outside the function.
2. pointers reduces length and complexity of the program and increase execution speed.
3. pointers are more efficient in handling data tables.
4. pointers are also more efficient in handling the data array.
5. pointers are used to create complex data structures.

such as linked lists, trees.

~~Ques~~ Q. what are the two operators used in pointers?

→ the two operators used in pointer are as follows:

a. Address of operator (&) → ampersand

Address of operator is the ampersand, i.e &.

It is a unary operator i.e. applied to a variable to assign its memory address.

b. Indirection or dereference operator (\*) → asterisk

It is denoted by "\*" sign. This unary operator access the values resides at a memory address.

Q. How do you declare a pointer.

→ pointer declaration:

data-type \* pointer-variable;

i.e. int \* p;

# Find the output of the following program.

1. Int main()

{

int b = 25;

// memory location of b is 1234

int \* p;

p = & b ;

printf("%d %d", &b, p);

return 0;

3

- Ⓐ 25 25 Ⓑ 1234 1234 Ⓒ 25 1234 Ⓓ 1234 25

2. int main()

{

int a=5;

int \*q;

q=&a;

printf("%d %d", a, \*q);

return 0;

}

- Ⓐ 5 5 Ⓑ 5 1 Ⓒ errors Ⓓ 0 5

3. int main()

{

int x=5;

// memory address of x is 2000

int \*p;

p=&x;

printf("%d %d", p, \*p);

return 0;

}

- Ⓐ 5 5 Ⓑ 5 2000 Ⓒ 2000 2000 Ⓓ 2000.5

4. int main()

{

int x = 5, y = 10;

int \*p = &x, \*q = &y;

\*q = 5;

printf("%d", \*p + \*q);

return 0;

}

Ans: 10