

RCS 151/251 CSP Lab Solutions

1. WAP that calculates the simple interest and compound interest. The principal, amount, rate of interest and time are entered through the keyboard.

```
#include<stdio.h>
#include<math.h>
void main()
{
    int p,r,t;
    float si,ci,amount;
    printf("Enter principle, rate and time\n");
    scanf("%d%d%d",&p,&r,&t);
    si=(p*r*t)/100.0;
    amount=p*pow((1+r/100.0),t);
    ci=amount-p;
    printf("Simple Interest=%f\n",si);
    printf("Compound Interest=%f\n",ci);
}
```

Output:

```
Enter principle, rate and time
1000
10
5
Simple Interest=500.000000
Compound Interest=610.510010
```

2. WAP to calculate the area and circumference of a circle.

```
/* A program to calculate the area and circumference of a circle */
#include<stdio.h>
#define PI 3.14
void main()
{
    float r, area, circum;
    printf("Enter radius of circle\n");
    scanf("%f",&r);
    area=2*PI*r;
    circum=PI*r*r;
    printf("Area is=%f\n",area);
    printf("Circumference=%f\n",circum);
}
```

Output:

```
Enter radius of circle
10
Area is=62.799999
Circumference=314.000000
```

3. WAP that accepts the temperature in centigrade and converts into fahrenheit using the formula $C/5=(F-32)/9$.

```
void main()
{
    float c,f;
    printf("Enter the tempertaure in centigrade\n");
    scanf("%f",&c);
    f=9.0/5.0*c+32.0;
    printf("Temperature in farenhiet is: %f\n",f);
}
```

Output:

```
Enter the tempertaure in centigrade
100
Temperature in farenhiet is: 212.000000
```

4. WAP that swaps values of two variables using a third variable.

```
#include<stdio.h>
void main()
{
    int a,b, temp;
    printf("Enter two values a and b\n");
    scanf("%d%d",&a,&b);
    temp=a;
    a=b;
    b=temp;
    printf("Values after swapping is\n");
    printf("a=%d\nb=%d\n",a,b);
}
```

Output:

```
Enter two values a and b
5
6
Values after swapping is
a=6
b=5
```

5. WAP that swaps values of two variables without using a third variable.

```
#include<stdio.h>
void main()
{
    int a,b;
    printf("Enter two values a and b\n");
    scanf("%d%d",&a,&b);
    a=a+b;
    b=a-b;
```

```

    a=a-b;
    printf("Values after swapping is\n");
    printf("a=%d\nb=%d\n",a,b);
}

```

Output:

```

Enter two values a and b
5
6
Values after swapping is
a=6
b=5

```

5. WAP to print Hello on the screen without using semicolon (;).

```

#include<stdio.h>
void main()
{
    if(printf("Hello \n"))
    {
    }
}

```

Output:

```

Hello

```

6. WAP that finds whether a given number is even or odd.

```

#include<stdio.h>
void main()
{
    int n;
    printf("Enter a number\n");
    scanf("%d",&n);
    if(n%2==0)
        printf("Number is even");
    else
        printf("Number is odd");
}

```

Output:

```

Enter a number
56
Number is even

```

7. WAP to find the greatest of three numbers.

```
#include<stdio.h>
void main()
{
    int a,b,c;
    printf("Enter three numbers\n");
    scanf("%d%d%d",&a,&b,&c);
    if(a>=b && a>=c)
        printf("%d is greatest\n",a);
    else if(b>=a && b>=c)
        printf("%d is greatest\n",b);
    else
        printf("%d is greatest\n",c);
}
```

Output:

```
Enter three numbers
8
9
1
9 is greatest
```

8. WAP that tells whether a given year is a leap year or not.

```
#include<stdio.h>
void main()
{
    int year;
    printf("Enter a year\n");
    scanf("%d",&year);
    if((year%4==0&&year%100!=0)||(year%400==0))
        printf("Leap year\n");
    else
        printf("Not Leap Year\n");
}
```

Output:

```
Enter a year
2001
Not Leap Year
```

9. WAP that accepts marks of five subjects and finds percentage and prints grades according to the following criteria:

between 90-100%-----Print 'A'
80-89%-----Print 'B'
60-79%-----Print 'C'
below 60%-----Print 'D'

```

#include<stdio.h>
void main()
{
    int sub1,sub2,sub3,sub4,sub5,total;
    float percent;
    printf("Enter marks of 5 subjects out of 100\n");
    scanf("%d%d%d%d%d",&sub1,&sub2,&sub3,&sub4,&sub5);
    total=sub1+sub2+sub3+sub4+sub5;
    percent=(float)total/5;
    printf("Percentage is %f\n",percent);
    if(percent>=90 && percent<=100)
        printf("Grade A\n");
    else if(percent>=80 && percent<=89)
        printf("Grade B\n");
    else if(percent>=60 && percent<=79)
        printf("Grade C\n");
    else if(percent<60)
        printf("Grade D\n");
}

```

Output:

```

Enter marks of 5 subjects out of 100
89
87
65
90
78
Percentage is 81.800003
Grade B

```

10. WAP that takes two operands and one operator from the user and perform the operation and prints the result by using switch statement.

```

#include<stdio.h>
void main()
{
    int a,b,c,add,mul,sub;
    float div;
    char o;
    printf("Enter two operands\n");
    scanf("%d%d",&a,&b);
    printf("Enter an operator (+, -, *, /) \n");
    o=getchar();/*To take single character input */
    switch(o)
    {
        case '+': add=a+b;
                printf("Addition is %d\n",add);
                break;
        case '-': sub=a-b;
                printf("Subtraction is %d\n", sub);
                break;
        case '*': mul=a*b;

```

```

        printf("Multiplication is %d\n",mul);
        break;
    case '/': div=(float)a/(float)b;
        printf("Division is %f\n",div);
        break;
    default: printf("Sorry you have entered wrong operator\n");
}
}

```

Output:

Enter two operands

9

8

Enter an operator (+, -, *, /)

*

Multiplication is 72

11. WAP to demonstrate the use of break and continue statements.

BREAK

/* A program to print 1 to n natural numbers. However if the value of number to be printed becomes greater than 75 then stop further execution

NOTE: break is used to come out of the loop before actual loop condition becomes false */

```

#include<stdio.h>
void main()
{
    int i,n;
    printf("Enter the value of n: ");
    scanf("%d",&n);
    i=1;
    while(i<=n)
    {
        if(i>75) /*checking whether the ith value is less than 75 */
            break;
        printf("%d ",i);
        i++;
    }
}

```

Output:

Enter the value of n: 6

1 2 3 4 5 6

CONTINUE:

/* Example of Continue (Note continue is used to skip a portion of the loop and jump to next iteration of the loop.

WAP to multiply n different values entered by the user. However if a user enters a negative value then skip it (i.e do not include it in multiplication */

```
#include<stdio.h>
void main()
{
    int n,i,value,prod=1;
    printf("Enter n\n");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        printf("Enter a value: ");
        scanf("%d",&value);
        if(value==0)
            continue;
        prod=prod*value;
    }
    printf("The product is %d\n",prod);
}
```

Output:

```
Enter n
5
Enter a value: 2
Enter a value: 2
Enter a value: 0
Enter a value: 3
Enter a value: 2
The product is 24
```

12. Write a program in C to print the following pattern

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

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

Output:

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

13. Write a program in C to print the following pattern

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

```
#include<stdio.h>
void main()
{
    int i,j;
    for(i=5;i>=1;i--)
    {
        for(j=1;j<=i;j++)
        {
            printf("*");
        }
        printf("\n");
    }
}
```

Output:

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

14. Write a program in C to print the following pattern

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

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



```

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

```

Output:

```

1
22
333
4444
55555

```

15. WAP to print following design:

```

A
B A
A B A
B A B A
A B A B A

```

```

#include<stdio.h>
int main()
{
    int i,j;
    for(i=1;i<=5;i++)
    {
        for(j=i;j>=1;j--)
        {
            if(j%2==0)
                printf("B ");
            else
                printf("A ");
        }
        printf("\n");
    }
    return 0;
}

```

Output:

```

A
B A
A B A

```

B A B A
A B A B A

16. WAP to print following design

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

```
#include<stdio.h>
int main()
{
    int i,j,k,space=4;
    for(i=1;i<=5;i++) // to control rows
    {
        for(j=1;j<=space;j++) //to control and display spaces
        {
            printf(" ");
        }
        space--; //To reduce the number of spaces to be printed
        for(k=1;k<=i;k++)// To control column and display *
        {
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

Output:

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

17. WAP to print the sum of the following series

1 -2 +3 -4+5 upto nth term

$1^2+3^2+5^2+\dots$ upto nth term

/* Sum of: 1 -2 +3 -4+5 upto nth term */

```
#include<stdio.h>
int main()
{
    int i,n,sum=0;
    printf("Enter the value of n: ");
```

```

scanf("%d",&n);
for(i=1;i<=n;i++)
{
    if(i%2==0)
        sum=sum-i;
    else
        sum=sum+i;
}
printf("The sum of series is %d\n",sum);
}

```

Output:

Enter the value of n: 5

The sum of series is 3

```

/* Sum of  $1^2+3^2+5^2+\dots$  upto nth term */
#include<stdio.h>
int main()
{
    int i,sum=0,n;
    printf("Enter total number of terms: ");
    scanf("%d",&n);
    for(i=1;i<(2*n);i=i+2)
    {
        sum=sum+(i*i);
    }
    printf("Sum of series is %d\n",sum);
    return 0;
}

```

Output:

Enter total number of terms: 3

Sum of series is 35

18. WAP to find the factorial of a given number.

```

#include<stdio.h>
void main()
{
    int i,n,fact=1;
    printf("Enter a number\n");
    scanf("%d",&n);
    for(i=n;i>=1;i--)
    {
        fact=fact*i;
    }
    printf("The factorial of %d is %d",n,fact);
}

```

Output:

Enter a number

5

The factorial of 5 is 120

19. WAP to print sum of even and odd numbers from 1 to N numbers.

```
#include<stdio.h>
void main()
{
    int i,n,sumEven=0,sumOdd=0;
    printf("Enter a number\n");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        if(i%2==0)
            sumEven=sumEven+i;
        else
            sumOdd=sumOdd+i;
    }
    printf("The sum of even numbers upto %d is %d\n",n,sumEven);
    printf("The sum of odd numbers upto %d is %d\n",n,sumOdd);
}
```

Output:

Enter a number

6

The sum of even numbers upto 6 is 12

The sum of odd numbers upto 6 is 9

20. WAP to check whether the entered number is prime or not.

```
#include<stdio.h>
void main()
{
    int i,n,count=0;
    printf("Enter a number\n");
    scanf("%d",&n);
    for(i=2;i<=n-1;i++)
    {
        if(n%i==0)
            count++;
    }
}
```

```

    }
    if(count==0)
        printf("%d is a prime number",n);
    else
        printf("%d is not a prime number",n);
}

```

Output:

Enter a number

7

7 is a prime number

21. WAP to find the sum of digits of the entered number.

```

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

```

Output:

Enter a number

152

The sum of digits is 8

22. WAP to find the reverse of a number.

```

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

```

```

    }
    printf("The sum of digits is %d",rev);
}

```

Output:

Enter a number

198

The sum of digits is 891

23. WAP to print armstrong numbers from 1 to 500.

Note: A number N is considered to be Armstrong number if the sum of its each individual digits raise to the power total number of digits in N is equal to the N itself.

E.g., 153 is Armstrong because $1^3+5^3+3^3$ is equal to 153.

```

#include<stdio.h>
#include<math.h>
#define TRUE 1
#define FALSE 0
int numOfDigits(int num);
int isArmstrong(int num);
int main()
{
    int i,v;
    for(i=1;i<=500;i++)
    {
        v=isArmstrong(i);
        if(v==TRUE)
            printf("%d is Armstrong number\n",i);
    }
    return 0;
}
int isArmstrong(int num)
{
    int d,r,sum=0,num2;
    num2=num;
    d=numOfDigits(num);
    while(num!=0)
    {
        r=num%10;
        sum=sum+pow(r,d);
        num=num/10;
    }
    if(sum==num2)
        return TRUE;
    else
        return FALSE;
}

```

```

int numOfDigits(int num)
{
    int nod=0;
    while(num!=0)
    {
        nod++;
        num=num/10;
    }
    return nod;
}

```

Output:

1 is Armstrong number
 2 is Armstrong number
 3 is Armstrong number
 4 is Armstrong number
 5 is Armstrong number
 6 is Armstrong number
 7 is Armstrong number
 8 is Armstrong number
 9 is Armstrong number
 153 is Armstrong number
 370 is Armstrong number
 371 is Armstrong number
 407 is Armstrong number

24. WAP to convert binary number into decimal number and vice versa.

/* A program to convert binary number into decimal number*/

```

#include<stdio.h>
#include<math.h>
void main()
{
    int n,bin,dec=0,c=0,d;
    printf("Enter a binary number\n");
    scanf("%d",&bin);
    while(bin!=0)
    {
        d=bin%10;
        dec=dec+d*pow(2,c);
        c++;
        bin=bin/10;
    }
    printf("Decimal equivalent is %d\n",dec);
}

```

Output:

Enter a binary number
 1111

Decimal equivalent is 15

```
/* A program to convert decimal number to binary number */
#include<stdio.h>
#include<math.h>
void main()
{
    int dec,bin=0,r,p=0;
    printf("Enter a decimal number\n");
    scanf("%d",&dec);
    while(dec!=0)
    {
        r=dec%2;
        bin=bin+r*pow(10,p); /* p will count the powers of 10 with which remainder is to be multiplied */
        p++;
        dec=dec/2;
    }
    printf("The binary equivalent is %d\n",bin);
}
```

Output:

Enter a decimal number

15

The binary equivalent is 1111

25. WAP that simply takes elements of the array from the user and finds the min and max of these elements.

```
#include<stdio.h>
void main()
{
    int a[100],N,max,min;
    printf("Enter total number of elements on array (<100)");
    scanf("%d",&N);
    printf("Enter array elements\n");
    for(i=0;i<N;i++)
        scanf("%d",&a[i]);
    max=a[0];
    min=a[0];
    for(i=1;i<n;i++)
    {
        if(a[i]>max)
            max=a[i];
        else if(a[i]<min)
            min=a[i];
    }
    printf("Maximum element=%d\n",max);
    printf("Minimum element=%d\n, min);
}
```


Output:

Enter total number of elements on array (<100) 5

Enter array elements 4 5 2 9 1

Maximum element=9

Minimum element=1

26.WAP that inputs two arrays and saves sum of corresponding elements of these arrays in a third array and prints them.

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

```
void addArray(int a[100], int b[100], int c[100], int N); //N is the actual size of array a,b and c
```

```
void main()
```

```
{
    int x[100],y[100],z[100],size,i;
    printf("Enter the size of the arrays:\n");
    scanf("%d",&size);
    printf("Enter first array\n");
    for(i=0;i<=size-1;i++)
        scanf("%d",&x[i]);
    printf("Enter second array\n");
    for(i=0;i<=size-1;i++)
        scanf("%d",&y[i]);
    addArray(x,y,z,size);
    printf("Resultant addition of arrays:\n");
    for(i=0;i<=size-1;i++)
        printf("%d\n",z[i]);
}
void addArray(int a[100], int b[100], int c[100],int N)
{
    int i;
    for(i=0;i<=N-1;i++)
        c[i]=a[i]+b[i];
}
```

Output:

Enter the size of the arrays:

5

Enter first array

2

3

6

8

4

Enter second array

6

5

4

7

3

Resultant addition of arrays:

8

8
10
15
7

**27. WAP to add and multiply two matrices of order nxn.
Add 2 matrices:**

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    int i,j,m,n,a[100][100],b[100][100],c[100][100];
    printf("Enter rows m and columns n of matrix a and b\n");
    scanf("%d%d",&m,&n);
    printf("Enter the elements of matrix a\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Enter the elements of matrix b\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
    /*Adding a and b in c*/
    for(i=0;i<m;i++)
    {
        /*Number of rows in c*/
        for(j=0;j<n;j++)
        {
            /*Number of columns in c*/
            c[i][j]=a[i][j]+b[i][j];
        }
    }
    printf("Matrix c is\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("%d\t",c[i][j]);
        }
    }
}
```

```

        }
        printf("\n");
    }
}

```

Output:

Enter rows m and columns n of matrix a and b

3

3

Enter the elements of matrix a

1 2 3

4 5 6

7 8 9

Enter the elements of matrix b

0 1 0

2 1 1

3 3 1

Matrix c is

1 3 3

6 6 7

10 11 10

Matrix Multiplication: [VERY IMPORTANT]

/* A program to multiply two matrices and b and store the result in third matrix c */

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

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

```
void main()
```

```
{
```

```
    int i,j,m,n,o,p,a[100][100],b[100][100],c[100][100],k,sum;
```

```
    printf("Enter rows m and columns n of first matrix\n");
```

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

```
    printf("Enter rows o and columns p of second matrix\n");
```

```
    scanf("%d%d",&o,&p);
```

```
    if(n!=o)
```

```
    {
```

```
        printf("Matrices can not be multiplied\n");
```

```
        exit(0);
```

```
    }
```

```
    printf("Enter the elements of matrix a\n");
```

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

```
    {
```

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

```
        {
```

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

```

    }
}
printf("Enter the elements of matrix b\n");
for(i=0;i<o;i++)
{
    for(j=0;j<p;j++)
    {
        scanf("%d",&b[i][j]);
    }
}
/*Multiplying a and b in c*/
for(i=0;i<m;i++)
{ /*Number of rows in c*/
    for(j=0;j<p;j++)
    { /*Number of columns in c*/
        sum=0;
        /*Adding the value using the formula */
        for(k=0;k<n;k++)
        {
            sum=sum+(a[i][k]*b[k][j]);
        }
        c[i][j]=sum;
    }
}
printf("Matrix c is\n");
for(i=0;i<m;i++)
{
    for(j=0;j<p;j++)
    {
        printf("%d\t",c[i][j]);
    }
    printf("\n");
}
}

```

Output:

```

Enter rows m and columns n of first matrix
2
3
Enter rows o and columns p of second matrix
3
2
Enter the elements of matrix a
1 2 3
3 2 1
Enter the elements of matrix b
2 2

```

2 2

2 2

Matrix c is

12 12

12 12

29. WAP to find the minimum and maximum element of the array by using functions.

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

```
int max(int a[100],int n);
```

```
int min(int a[100], int n);
```

```
void main()
```

```
{
```

```
    int arr[100],size,i,maxVal,minVal;
```

```
    printf("Enter the size of the array\n");
```

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

```
    printf("Enter array elements\n");
```

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

```
    {
```

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

```
    }
```

```
    maxVal=max(arr,size);/*Function call: array arr and variable size are passed as argument */
```

```
    minVal=min(arr,size);
```

```
    printf("The maximum element is %d\n",maxVal);
```

```
    printf("The minimum element is %d\n",minVal);
```

```
}
```

```
int max(int a[100], int n)
```

```
{
```

```
    int i,highest;
```

```
    highest=a[0];
```

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

```
    {
```

```
        if(a[i]>highest)
```

```
            highest=a[i];
```

```
    }
```

```
    return highest;
```

```
}
```

```
int min(int a[100], int n)
```

```
{
```

```
    int i,lowest;
```

```
    lowest=a[0];
```

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

```
    {
```

```
        if(a[i]<lowest)
```

```
            lowest=a[i];
```

```
    }
```

```
    return lowest;
```

```
}
```

Output:

Enter the size of the array

5

Enter array elements

5

3

9

8

1

The maximum element is 9

The minimum element is 1

30. WAP to search an element in an array using linear search by using function.

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

```
#define NOT_FOUND -1
```

```
int linearSearch(int a[100],int N,int item);
```

```
int main()
```

```
{
```

```
    int arr[100], size,i,search_item,index;
```

```
    printf("Enter size of array: ");
```

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

```
    printf("Enter array elements:\n");
```

```
    for(i=0;i<=size-1;i++)
```

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

```
    printf("Enter item to search: ");
```

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

```
    index=linearSearch(arr,size,search_item);
```

```
    if(index==NOT_FOUND)
```

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

```
    else
```

```
        printf("Item found at %d index\n",index);
```

```
    return 0;
```

```
}
```

```
int linearSearch(int a[100],int N,int item)
```

```
{
```

```
    int i;
```

```
    for(i=0;i<=N-1;i++)
```

```
    {
```

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

```
            return i;
```

```
    }
```

```
    return NOT_FOUND;
```

```
}
```

Output:

Enter size of array: 5

Enter array elements:

8

9

12

5

19

Enter item to search: 5

Item found at 3 index

31. WAP to sort the elements of the array in ascending order using bubble sort technique by using function.

```
/* Bubble sort WITHOUT USING FUNCTIONS */
```

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

```
void main()
```

```
{
```

```
    int i,j,l,n,temp,a[100];
```

```
    printf("Enter number of elements\n");
```

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

```
    printf("Enter the array\n");
```

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

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

```
    //Bubble sort begins
```

```
    for(l=n-1;l>0;l--)
```

```
    {
```

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

```
        {
```

```
            if(a[j]>a[j+1]) /* for sorting in descending order change > to < */
```

```
            {
```

```
                temp=a[j];
```

```
                a[j]=a[j+1];
```

```
                a[j+1]=temp;
```

```
            }
```

```
        }
```

```
    }
```

```
    printf("Array after sorting is\n");
```

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

```
        printf("%d\n",a[i]);
```

```
}
```

Output:

Enter number of elements

5

Enter the array

3

2
1
8
6
Array after sorting is
1
2
3
6
8

/* Bubble sort USING FUNCTIONS

NOTE: logic will remain the same. However we have to perform sorting in a separate function. We will pass the array and its size as arguments to a bubble_sort() function.

Since arrays are passed by reference so any changes made to the passed array in the CALLED function will be reflected back in the array in CALLING function. Also an array can never be returned from the CALLED function back to CALLING function */

```
#include<stdio.h>
void bubble_sort(int a[100],int n);
void main()
{
    int i,size,arr[100];
    printf("Enter number of elements\n");
    scanf("%d",&size);
    printf("Enter the array\n");
    for(i=0;i<size;i++)
        scanf("%d",&arr[i]);
    bubble_sort(arr,size); /* Function Call: Array arr and variable size are passed as arguments */
    printf("Array after sorting is\n");
    for(i=0;i<size;i++)
        printf("%d\n",arr[i]);
}
void bubble_sort(int a[100], int n)
{
    int l,j,temp;
    //Bubble sort begins
    for(l=n-1;l>0;l--)
    {
        for(j=0;j<l;j++)
        {
            if(a[j]>a[j+1])/* for sorting in descending order change > to < */
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
}
```



```

    }
}
}

```

Output:

Enter number of elements

3

Enter the array

9

0

17

Array after sorting is

0

9

17

32. WAP to implement strlen(), strcat(), strcmp(), strcpy() using user defined functions.

A string is a group of characters. Strings in C are just array of characters terminated by a '\0' (NULL) character. The NULL character ('\0') is automatically added by the compiler after the string has been input or initialized.

strlen(string_name)- This function returns the total number of characters in a string without considering the NULL character.

Implementing strlen() without creating user-defined functions:

```

#include<stdio.h>
void main()
{
    char str[100];
    int i,len=0;
    printf("Enter string: ");
    gets(str);
    for(i=0;str[i]!='\0';i++)
        len++;
    printf("Length of string = %d\n",len);
}

```

Implementing strlen() by creating user-defined functions: We will pass the string as an argument to a function. Since string is basically an array of characters in C so it will be passed by reference. Note: ANY ARRAY INCLUDING STRING CAN NEVER BE RETURNED FROM A FUNCTION

```

#include<stdio.h>
int stringLength(char str[100]);
void main()

```

```

{
    char s[100];
    int len;
    printf("Enter a string: ");
    gets(s);
    len=stringLength(s);
    printf("Length of string is %d\n",len);
}
int stringLength(char str[100])
{
    int i,length=0;
    for(i=0;str[i]!='\0';i++)
        length++;
    return length;
}

```

Output:

Enter a string: lucky rajput
Length of string is 12

strcat(dest,src)- This function joins the source string i.e., src at the end of destination string i.e., dest

Implementing strcat() without using creating functions:

```

#include<stdio.h>
#include<string.h>
void main()
{
    char src[100],dest[200];
    int i,j;
    printf("Enter source string: ");
    gets(src);
    printf("Enter destination string: ");
    gets(dest);
    i=0;
    j=strlen(dest);
    while(src[i]!='\0')
    {
        dest[j]=src[i];
        i++;
        j++;
    }
    dest[j]='\0'; /*We have copied all the characters of the source at the end of destination except
the '\0' character. So lastly we add '\0' at the end of destination */
    printf("Destination string concatenated with source string = ");
    puts(dest);
}

```

```
}
```

Implementing strcat() by creating user-defined functions:

```
#include<stdio.h>
#include<string.h>
void stringCat(char dest[100],char src[200]);
void main()
{
    char s[100],d[200];
    printf("Enter source: ");
    gets(s);
    printf("Enter destination: ");
    gets(d);
    stringCat(d,s); /*string d and s are passed by reference and they will be recieved in dest and
src so any changes made to dest and src will automatically reflected back in d and s */
    printf("Strings after concatenation\n");
    printf("Source: ");
    puts(s);
    printf("Destination: ");
    puts(d);
}
void stringCat(char dest[100],char src[200])
{
    int i,j;
    i=0;
    j=strlen(dest);
    while(src[i]!='\0')
    {
        dest[j]=src[i];
        i++;
        j++;
    }
    dest[j]='\0'; /*We have copied all the characters of the source at the end of destination except
the '\0' character. So lastly we add '\0' at the end of destination */
}
```

Output:

```
Enter source: Hitesh
Enter destination: Ahuja
Strings after concatenation
Source: Hitesh
Destination: AhujaHitesh
```

strcpy(dest,src)- This function copies the contents of source string in destination string.

Implementing strcpy without creating user defined function:

```
#include<stdio.h>
void main()
{
    char src[100],dest[100];
    int i,j;
    printf("Enter source string: ");
    gets(src);
    i=0;
    j=0;
    while(src[i]!='\0')
    {
        dest[j]=src[i];
        i++;
        j++;
    }
    dest[j]='\0'; /*We have copied all the characters of the source in destination except the '\0'
character. So lastly we add '\0' at the end of destination */
    printf("Source string copied into destination string: ");
    puts(dest);
}
```

Implementing strcpy by creating user defined function:

```
#include<stdio.h>
void stringCopy(char dest[100], char src[100]);
void main()
{
    char s[100],d[100];
    printf("Enter source string: ");
    gets(s);
    stringCopy(d,s); /*string d and s are passed by reference and they will be recieved in dest and
src so any changes made to dest and src will automatically reflected back in d and s */
    printf("Source string copied into destination string: ");
    puts(d);
}
void stringCopy(char dest[100], char src[100])
{
    int i,j;
    i=0;
    j=0;
    while(src[i]!='\0')
    {
        dest[j]=src[i];
        i++;
    }
}
```

```

        j++;
    }
    dest[j]='\0'; /*We have copied all the characters of the source in destination except the '\0'
    character. So lastly we add '\0' at the end of destination */
}

```

Output:

Enter source string: Ghaziabad

Source string copied into destination string: Ghaziabad

33. WAP to generate fibnocii series (0 1 1 2 3 5 8 13...) using recursive function.

```

#include<stdio.h>
int fib(int k);
void main()
{
    int i,n,val;
    printf("Enter number of terms upto which you want to generate fibonacci series: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    { /* this loop will print all the ith fibonacci terms from 1 to n*/
        val=fib(i);
        printf("%d\t",val);
    }
}
int fib(int k)/*This function will return kth fibonacci term */
{
    if(k==1) /* base case 1*/
        return 0;
    else if(k==2) /* base case 2*/
        return 1;
    else /* recursive case*/
        return(fib(k-1)+fib(k-2));
}

```

Output:

Enter number of terms upto which you want to generate fibonacci series: 5

0 1 1 2 3

34. Write a recursive call to calculate X^Y using recursive function.

```

#include<stdio.h>
int xPowerY(int x, int y);
void main()
{
    int x1,y1,p;
    printf("Enter the number: ");
    scanf("%d",&x1);
    printf("Enter the power: ");

```

```

scanf("%d",&y1);
p=xPowerY(x1,y1);
printf("X to the power y is %d\n",p);
}
int xPowerY(int x,int y)
{
    if(y==0) /* base case */
        return 1;
    else /* recursive case */
        return x*xPowerY(x,y-1);
}

```

Output:

Enter the number: 4
Enter the power: 3
X to the power y is 64

35. WAP to calculate the sum and multiplication of two complex numbers by making the use of structures.

```

#include<stdio.h>
struct complex
{
    int real; /*to store real part */
    int img; /*to store imaginary part */
} x,y,add,mul;
void main()
{
    printf("Enter the real part of first number: ");
    scanf("%d",&x.real);
    printf("Enter the imaginary part of first number: ");
    scanf("%d",&x.img);
    printf("First number entered is %d+%di\n",x.real,x.img);
    printf("Enter the real part of second number: ");
    scanf("%d",&y.real);
    printf("Enter the imaginary part of second number: ");
    scanf("%d",&y.img);
    printf("Second number entered is %d+%di\n",y.real,y.img);
    add.real=x.real+y.real;
    add.img=x.img+y.img;
    printf("Addition is %d+%di\n",add.real,add.img);
    mul.real=x.real*y.real-x.img*y.img;
    mul.img=x.real*y.img+x.img*y.real;
    printf("Multiplication is %d+%di",mul.real,mul.img);
}

```

Output:

Enter the real part of first number: 2
Enter the imaginary part of first number: 3
First number entered is 2+3i
Enter the real part of second number: 4
Enter the imaginary part of second number: 3
Second number entered is 4+3i
Addition is 6+6i
Multiplication is -1+18i

36. Create a structure student having following members: Roll_number, Name, Department, Course, Year_of_joining. Assume there are 100 students. WAP to print names of all students who joined in a particular year.

```
#include<stdio.h>

struct student
{
    int roll;
    char name[50];
    char dept[10];
    char course[10];
    int year;
}s[100]; /* s is an array of 100 elements and each element will be a student structure*/
void main()
{
    int i,y;
    for(i=0;i<100;i++)
    { /* Taking input in array of structures */
        printf("Enter details of student: %d\n",i+1);
        printf("Enter roll number\n");
        scanf("%d",&s[i].roll);
        printf("Enter name\n");
        gets(s[i].name);
        printf("Enter Department \n");
        gets(s[i].dept);
        printf("Enter course\n");
        gets(s[i].course);
        printf("Enter year of admission\n");
        scanf("%d",&s[i].year);
        printf("\n");
    }
    printf("Enter year of joining\n");
    scanf("%d",&y);
    for(i=0;i<100;i++)
    { /* performing linear search on array of structures */
```

```

        if(s[i].year==y)
        {
            printf("Roll number: ");
            printf("%d\n",s[i].roll);
            printf("Name: ");
            puts(s[i].name);
            printf("Department: ");
            puts(s[i].dept);
            printf("Course: ");
            puts(s[i].course);
            printf("Year of admission: ");
            printf("%d\n",s[i].year);
            printf("\n");
        }
    }
}

```

Output:

Enter details of student: 1

Enter roll number

1

Enter name

Priyank

Enter Department

IT

Enter course

BTech

Enter year of admission

2014

Enter details of student: 2

Enter roll number

2

Enter name

Hitesh

Enter Department

CSE

Enter course

BTech

Enter year of admission

2015

Enter details of student: 3

Enter roll number

56

Enter name

Mohit
Enter Department
Computers
Enter course
MCA
Enter year of admission
2009
.
.
.
Enter year of joining
2015
Roll number: 2
Name: Hitesh
Department: CSE
Course: BTech
Year of admission: 2015

37. WAP to swap two elements using the concept of pointers.

** This is an example of call/pass by reference */*

```
#include<stdio.h>
void swap(int *x, int *y);
void main()
{
    int a,b;
    printf("Enter the value of a and b\n");
    scanf("%d%d",&a,&b);
    printf("Values before swapping\n");
    printf("a=%d b=%d\n",a,b);
    swap(&a,&b); /* Pass by reference */
    printf("Values after swapping\n");
    printf("a=%d b=%d\n",a,b);
}
void swap(int *x, int *y)
{
    int t;
    t=*x;
    *x=*y;
    *y=t;
}
```

Output:

Enter the value of a and b
5

8

Values before swapping

a=5 b=8

Values after swapping

a=8 b=5

38. WAP to check whether a string is palindrome or not using the concept of pointers.

/* Any object is palindrome if its reversed form is equal to original form

e.g. 12321 is palindrome

MADAM is palindrome etc */

/* NOTE: A string is just an array of characters terminated by a \0 (NULL) character */

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

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

```
void main()
```

```
{
```

```
    char str[10],*ptr1,*ptr2;
```

```
    int l,i;
```

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

```
    gets(str);
```

```
    l=strlen(str);
```

```
    ptr1=str; /* ptr1 is pointing to the first character of str */
```

```
    ptr2=&str[l-1]; /* ptr2 is pointing to the (l-1)th character of str */
```

```
    while(*ptr1!='\0')
```

```
    {
```

```
        if(*ptr1!=*ptr2)
```

```
        { /* If string is not a palindrome then display a message and exit the program */
```

```
            printf("Not a palindrome\n");
```

```
            exit(1);
```

```
        }
```

```
        ptr1++;
```

```
        ptr2--;
```

```
    }
```

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

```
}
```

Output:

Enter a string:

madam

Palindrome

39. WAP to compare the contents of two files and determine whether they are same or not.

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
    FILE *fp1,*fp2;
    char c1,c2;
    fp1=fopen("file1","r");
    fp2=fopen("file2","r");
    c1=getc(fp1); /*Read first character from file1 */
    c2=getc(fp2); /*Read first character from file2 */
    while(c1!=EOF&& c2!=EOF)
    { /* This loop will only execute till ANY ONE of the file has been completely read */
        if(c1!=c2)
        {
            printf("Files are not same\n");
            exit(0);
        }
        c1=getc(fp1); /*Read next character from file1 */
        c2=getc(fp2); /*Read next character from file2 */
    }
    /* Now it may be possible that one file is smaller than other file. However its contents are
    subset of larger file. In that case also, we will reach at this point but files will not be same. To
    check this specific condition, we ensure that both the file pointer contain EOF which means
    both will be pointing to the end of the files then one file cannot be subset of another file */
    if(c1==EOF && c2==EOF)
        printf("Files are same\n");
    fclose(fp1);
    fclose(fp2);
}
```

Contents of file1:

This is a test file

Contents of file2:

This is a test file

Output:

Files are same

40. Create a file integer and store integers from 1 to 20. Now create two files even and odd. Read the integer file and store even numbers in even file and odd numbers in odd file. Finally display the contents of all the three files.

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

```

{
    FILE *fp,*fp1,*fp2;
    int n,c,i;
    fp=fopen("num.dat","w+");
    fp1=fopen("even.dat","w+");
    fp2=fopen("odd.dat","w+");
    for(i=1;i<=20;i++)
    {
        fprintf(fp,"%d\n",i);
    }
    rewind(fp);
    fscanf(fp,"%d",&n);
    while(feof(fp)==0)
    {
        if(n%2==0)
            fprintf(fp1,"%d ",n);
        else
            fprintf(fp2,"%d ",n);
        fscanf(fp,"%d",&n);
    }
    rewind(fp);
    rewind(fp1);
    rewind(fp2);
    printf("\nContents of the first file (num.dat)\n");
    fscanf(fp,"%d",&n);
    while(feof(fp)==0)
    {
        printf("%d ",n);
        fscanf(fp,"%d",&n);
    }
    printf("\nContents of the second file (even.dat)\n");
    fscanf(fp1,"%d",&n);/
    while(feof(fp1)==0)
    {
        printf("%d ",n);
        fscanf(fp1,"%d",&n);
    }
    printf("\nContents of the third file (odd.dat)\n");
    fscanf(fp2,"%d",&n);
    while(feof(fp2)==0)
    {
        printf("%d ",n);
        fscanf(fp2,"%d",&n);
    }
    fclose(fp);
    fclose(fp1);
}

```

```

        fclose(fp2);
    }
}

```

Source Code Explained:

```

#include<stdio.h>
void main()
{
    FILE *fp,*fp1,*fp2;
    int n,c,i;
    fp=fopen("num.dat","w+");/* To read and write both*/
    fp1=fopen("even.dat","w+");
    fp2=fopen("odd.dat","w+");
    /* First we write numbers from 1 to 20 to num.dat */
    for(i=1;i<=20;i++)
    {
        fprintf(fp,"%d\n",i);
    }
    /*Now the file pointer fp is pointing to the end of the file*/
    rewind(fp); /*To rewind the file pointer of first file so that it again starts pointing to the first element of the file */
    /* Now we write even numbers of num.dat to even.dat and odd numbers of num.dat to odd.dat */
    fscanf(fp,"%d",&n); /* Read first number from file (num.dat)*/
    while(feof(fp)==0) /* feof() returns 0 untill the file pointer is not pointing to the end of the file */
    {
        if(n%2==0)
            fprintf(fp1,"%d ",n); /*Write even numbers to even.dat file using file pointer fp1 */
        else
            fprintf(fp2,"%d ",n); /* Write odd numbers to odd.dat file using file pointer fp2 */
        fscanf(fp,"%d",&n); /* Read next number from file (num.dat)*/
    }
    /* Now we have to display the contents of all three files */
    /* The file pointers of all three files will be pointing to the end of the file. We have again set
    them to the starting of the file using rewind function */
    rewind(fp);
    rewind(fp1);
    rewind(fp2);
    /*Displaying the contents of the first file */
    printf("\nContents of the first file (num.dat)\n");
    fscanf(fp,"%d",&n); /* Read first number from file*/
    while(feof(fp)==0) /* feof() returns 0 untill the file pointer is not pointing to the end of the file */
    {
        printf("%d ",n);
        fscanf(fp,"%d",&n); /* Read next number from file */
    }
    printf("\nContents of the second file (even.dat)\n");
    fscanf(fp1,"%d",&n);/* Read first number from file*/
    while(feof(fp1)==0) /*feof() returns 0 untill the file pointer is not pointing to the end of the file */

```

```

{
    printf("%d ",n);
    fscanf(fp1,"%d",&n); /* Read next number from file */
}
printf("\nContents of the third file (odd.dat)\n");
fscanf(fp2,"%d",&n);/* Read first number from file*/
while(feof(fp2)==0) /* feof() returns 0 untill the file pointer is not pointing to the end of the file */
{
    printf("%d ",n);
    fscanf(fp2,"%d",&n); /* Read next number from file */
}
fclose(fp);
fclose(fp1);
fclose(fp2);
}

```

Output:

Contents of the first file (num.dat)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Contents of the second file (even.dat)

2 4 6 8 10 12 14 16 18 20

Contents of the third file (odd.dat)

1 3 5 7 9 11 13 15 17 19