| S.No: 7 | Exp. Name: ***Design a C program which determines factorial of numbers*** | Date: 2023-04-22 |
|---|---|---|





**Aim:**

Design a C program which determines the numbers whose factorial values are between(including) minimum and maximum values.

**For example:**The value of 6! is 720, 7! is 5040 and 8! is 40320. The factorial of 7 (5040) exists between the given limits.

**Constraints:**1 <= min,max <= 103

**Instruction:**Your input and output layout must match exactly with the layout of the visible sample test cases.

**Source Code:**

factorial.c

```
#include<stdio.h>
void main()
{
        int fact=1,i,max,min,x=1;
        printf("Min: ");
        scanf("%d",&min);
        printf("Max: ");
        scanf("%d",&max);
        printf("Values: ");
        for(i=1;i<=max;i++)
        {
                fact=fact*i;
                if(fact>=min&&fact<=max)
                {
                                if(x==1)
                                {
                                        printf("%d ",i);
                                        x=0;
                                }
                                else
                                printf("%d ",i);
                        }
                }
                printf("\n");
        }
```





## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Min: |
| 5 |
| Max: |
| 10 |

| Values: 3 |
|---|

| Test Case - 2 |
|---|
| **User Output** |
| Min: |
| 5 |
| Max: |
| 29 |
| Values: 3 4 |

| S.No: 9 | Exp. Name: ***Design a C program to print the sequence of numbers in which each number is the sum of the three most recent predecessors*** | **Date: 2023-04-22** |
|---|---|---|

**Aim:**

Design a C program to print the sequence of numbers in which each number is the sum of the three most recent predecessors. Assume first three numbers as **0**, **1**, and **1**, print the result as shown in the example.

**Sample Input and Output:**

```
Enter the number of terms: 7
First 7 terms in the series are:
0
1
1
2
4
7
13
```

**Source Code:**

first.c

```
#include<stdio.h>
int main()
{
        int t1=0,t2=1,t3=1,t4,n,i;
        printf("Enter the number of terms: ");
        scanf("%d",&n);
        printf("First %d terms in the series are:",n);
        printf("\n%d\n%d\n%d\n",t1,t2,t3);
        for(i=4;i<=n;i++)
        {
                t4=t1+t2+t3;
                printf("%d\n",t4);
                t1=t2;
                t2=t3;
                t3=t4;
        }
        return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the number of terms: |
| 5 |
| First 5 terms in the series are: |
| 0 |
| 1 |
| 1 |
| 2 |

| 4 |
|---|

| Test Case - 2 |
|---|
| **User Output** |
| Enter the number of terms: |
| 7 |
| First 7 terms in the series are: |
| 0 |
| 1 |
| 1 |
| 2 |
| 4 |
| 7 |
| 13 |

| Test Case - 3 |
|---|
| **User Output** |
| Enter the number of terms: |
| 13 |
| First 13 terms in the series are: |
| 0 |
| 1 |
| 1 |
| 2 |
| 4 |
| 7 |
| 13 |
| 24 |
| 44 |
| 81 |
| 149 |
| 274 |
| 504 |

| S.No: 17 | Exp. Name: ***Write a C program to implement the string manipulation operations by using library functions.*** | Date: 2023-04-22 |
|---|---|---|

**Aim:**

Write a program to implement the string manipulation operations by using string library functions.

At the time of execution, the program should print the message on the console as:

```
Enter two strings :
```

For example, if the user gives the input as:

```
Enter two strings : Ram Laxman
```

then the program should print the result as:

```
The length of Ram : 3
The copied string of Ram : Ram
Ram is greater than Laxman
The concatenated string : RamLaxman
```

**Note:** Do use the printf() function with a newline character (\n) at the end.

**Source Code:**

str.c

```
#include<stdio.h>
#include<string.h>
void main()
{
        char str1[100], str2[100];
        int len;
        printf("Enter two strings : ");
        scanf("%s %s",str1,str2);
        len= strlen(str1);
        printf("The length of %s : %d\n",str1,len);
        printf("The copied string of %s : %s\n",str1,strcpy(str1,str1));
        int i=strcmp(str1,str2);
        if(i==0)
        {
                printf("Both strings are equal\n",str1,str2);
        }
        else if(i>0)
        {
                printf("%s is greater than %s\n",str1,str2);
        }
        else
        {
                printf("%s is less than %s\n",str1,str2);
        }
        printf("The concatenated string : %s\n",strcat(str1,str2));
        printf("\n");
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|

| User Output |
| --- |
| Enter two strings : |
| Ram Laxman |
| The length of Ram : 3 |
| The copied string of Ram : Ram |
| Ram is greater than Laxman |
| The concatenated string : RamLaxman |

| Test Case - 2 |
| --- |
| **User Output** |
| Enter two strings : |
| Faculty Bird |
| The length of Faculty : 7 |
| The copied string of Faculty : Faculty |
| Faculty is greater than Bird |
| The concatenated string : FacultyBird |

| S.No: 19 | Exp. Name: ***Write a C program to sort elements using insertion sort*** | Date: 2023-04-22 |
|---|---|---|





**Aim:**

Write a program to sort the elements in ascending order with insertion sort technique using functions.

At the time of execution, the program should print the message on the console as:

```
Enter n value :
```

For example, if the user gives the input as:

```
Enter n value : 3
```

Next, the program should print the message on the console as:

```
Enter 3 elements :
```

if the user gives the input as:

```
Enter 3 elements : 45 67 34
```

then the program should print the result as:

```
Elements before sorting : 45 67 34
Elements after sorting : 34 45 67
```

**Note:** Do use printf() with '\n' at the end of output.

**Source Code:**

sort.c

```c
#include<stdio.h>
void insertion_sort(int [], int);
void read(int [], int);
void display(int [], int);
void main()
{
        int a[20],n,i;
        printf("Enter n value : ");
        scanf("%d",&n);
        read(a,n);
        printf("Elements before sorting : ");
        display(a,n);
        insertion_sort(a,n);
        printf("Elements after sorting : ");
        display(a,n);
}
void insertion_sort(int a[],int n)
{
        int i,j,k;
        for(i=1;i<n;i++)
        {
                k=a[i];
                j=i-1;
                while(j>=0&&a[j]>k)
                {
                        a[j+1]=a[j];
                        j=j-1;
                }
                a[j+1]=k;
        }
}
void read(int a[],int n)
{
        int i;
        printf("Enter %d elements : ",n);
        for(i=0;i<n;i++)
        scanf("%d",&a[i]);
}
void display(int a[],int n)
{
        int i;
        for(i=0;i<n;i++)
        printf("%d ",a[i]);
        printf("\n");
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
| --- |
| **User Output** |
| Enter n value : |
| 3 |

```
45 67 34
Elements before sorting : 45 67 34
Elements after sorting : 34 45 67
```

| S.No: 20 | Exp. Name: ***Write a C program to sort the list of numbers using bubble sort and functions*** | Date: 2023-04-22 |
|---|---|---|





**Aim:**

Write a program to sort the elements in descending order with bubble sort technique using functions.

At the time of execution, the program should print the message on the console as:

```
Enter n value :
```

For example, if the user gives the input as:

```
Enter n value : 3
```

Next, the program should print the message on the console as:

```
Enter 3 elements :
```

if the user gives the input as:

```
Enter 3 elements : 45 67 34
```

then the program should print the result as:

```
Elements before sorting : 45 67 34
Elements after sorting : 67 45 34
```

**Note:** Write the functions read(), bubbleSort() and display() in sorta.c.

**Source Code:**

sort.c

```c
#include <stdio.h>
void bubbleSort(int [],int);
void read(int [],int);
void display(int[],int);
void main()
{
        int a[20], n, i;
        printf("Enter n value : ");
        scanf("%d", &n);
        read(a, n);
        printf("Elements before sorting : ");
        display(a, n);
        bubbleSort(a, n);
        printf("Elements after sorting : ");
        display(a, n);
}
void read(int a[],int n)
{
        int i;
        printf("Enter %d elements : ",n);
        for(i=0;i<n;i++)
        scanf("%d",&a[i]);
}
void display(int a[],int n)
{
        int i;
        for(i=0;i<n;i++)
        printf("%d ",a[i]);
        printf("\n");
}
void bubbleSort(int a[],int n)
{
        int i,j,temp;
        for(i=0;i<n-1;i++)
        {
                for(j=i+1;j<n;j++)
                {
                        if(a[j]>a[i])
                        {
                                temp=a[i];
                                a[i]=a[j];
                                a[j]=temp;
                        }
                }
        }
}
```

sorta.c

```
Type here...
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
| --- |
| **User Output** |
| Enter n value : |
| 3 |
| Enter 3 elements : |
| 4 6 8 |
| Elements before sorting : 4 6 8 |
| Elements after sorting : 8 6 4 |

| Test Case - 2 |
| --- |
| **User Output** |
| Enter n value : |
| 5 |
| Enter 5 elements : |
| 34 56 71 26 17 |
| Elements before sorting : 34 56 71 26 17 |
| Elements after sorting : 71 56 34 26 17 |

| S.No: 21 | Exp. Name: ***Write a C program to sort the elements using selection sort - largest element method technique*** | Date: 2023-04-22 |
|---|---|---|









**Aim:**

Write a program to sort the given array elements using selection sort largest element method.

At the time of execution, the program should print the message on the console as:

```
Enter value of n :
```

For example, if the user gives the input as:

```
Enter value of n : 3
```

Next, the program should print the messages one by one on the console as:

```
Enter element for a[0] :
Enter element for a[1] :
Enter element for a[2] :
```

if the user gives the input as:

```
Enter element for a[0] : 22
Enter element for a[1] : 33
Enter element for a[2] : 12
```

then the program should print the result as:

```
Before sorting the elements in the array are
Value of a[0] = 22
Value of a[1] = 33
Value of a[2] = 12
After sorting the elements in the array are
Value of a[0] = 12
Value of a[1] = 22
Value of a[2] = 33
```

Fill in the missing code so that it produces the desired result.

**Source Code:**

array.c

```c
#include<stdio.h>
void main()
{
        int a[20],i,n,j,max,temp=0;
        printf("Enter value of n : ");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
                printf("Enter element for a[%d] : ",i);
                scanf("%d",&a[i]);
        }
        printf("Before sorting the elements in the array are\n");
        for(i=0;i<n;i++)
        printf("Value of a[%d] = %d\n",i,a[i]);
        for(i=n-1;i>0;i--)
        {
                max=j;
                for(j=1;j>=0;j--)
                {
                        if(a[j]>=a[max])
                        max=j;
                }
                temp=a[i];
                a[i]=a[max];
                a[max]=temp;
        }
        printf("After sorting the elements in the array are\n");
        for(i=0;i<n;i++)
        {
                printf("Value of a[%d] = %d\n",i,a[i]);
        }
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter value of n : |
| 3 |
| Enter element for a[0] : |
| 15 68 48 |
| Enter element for a[1] : Enter element for a[2] : Before sorting the elements in the array are |
| Value of a[0] = 15 |
| Value of a[1] = 68 |
| Value of a[2] = 48 |
| After sorting the elements in the array are |
| Value of a[0] = 15 |
| Value of a[1] = 48 |
| Value of a[2] = 68 |