S.No: 11	Exp. Name: ***Develop an algorithm which computes the all the factors between 1 to 100 for a given number and implement it using C.***	Date:2023-04-02

ID: 224G1A0557

Aim:

Develop an algorithm which computes the all the factors between 1 to 100 for a given number and implement it using C.

Sample input output

Sample input output -1:

```
Enter a number: 23
Factors between 1 and 100 are: 1        23
```

Sample input output -2:

```
Enter a number: 234
Factors between 1 and 100 are: 1        2       3       6       9       13      18      26
```

Sample input output -3:

```
Enter a number: 5
Factors between 1 and 100 are: 1        5
```

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

Source Code:

factors100.c

```
#include<stdio.h>
main()
{
   int i,n;
   printf("Enter a number: ");
   scanf("%d",&n);
   printf("Factors between 1 and 100 are: ");
   for(i=1;i<=100;i++)
   {
      if(n%i==0)
      printf("%d\t",i);
   }
   printf("\n");
   return 0;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter a number: 45
Factors between 1 and 100 are: 1 3 5 9 15 45

2022-2026-CSE-A

Srinivasa Ramanujan Institute of Technology

S.No: 12	Exp. Name: ***Construct an algorithm which computes the sum of the factorials of numbers between m and n***	Date:2023-03-31

Aim:

Construct an algorithm which computes the sum of the factorials of numbers between m and n

Constraints:

m < n

Sample input output

Sample input output -1:

```
Enter m value: 3
Enter n value: 1
m value should be less than n
```

Sample input output -2:

```
Enter m value: 4
Enter n value: 6
Sum of factorials of numbers between 4 and 6 is 864
```

Sample input output -3:

```
Enter m value: 10
Enter n value: 13
Sum of factorials of numbers between 10 and 13 is 6749568000
```

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

Note: Use an appropriate data type for the variable storing the sum to accommodate large factorial values.

Source Code:

fact.c

```
#include<stdio.h>
int main()
{
   long int m,n,k,i,fact=1,sum=0;
   printf("Enter m value: ");
   scanf("%ld",&m);
   printf("Enter n value: ");
   scanf("%ld",&n);
   if(m<n)
   {
      printf("Sum of factorials of numbers between %ld and %ld is ",m,n);
      for(k=m;k<=n;k++)
      {
         fact=1;
         for(i=k;i>=1;i--)
      {
         fact=fact*i;
```

```
        }
        sum=sum+fact;
    }
    printf("%ld\n",sum);
    }
    else
    printf("m value should be less than n\n");
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter m value: 10
Enter n value: 13
Sum of factorials of numbers between 10 and 13 is 6749568000

Test Case - 2
User Output
Enter m value: 3
Enter n value: 1
m value should be less than n

ID: 224G1A0557
2022-2026-CSE-A
Srinivasa Ramanujan Institute of Technology

S.No: 13	Exp. Name: ***Write a C program to display the elements of an array in reverse order***	Date:2023-04-02

Aim:

Write a program to **print** the given integer elements of an array (with max size 10) in reverse order.

At the time of execution, the program should print the message on the console as:
Enter size of the array :
For example, if the user gives the **input** as:

```
Enter size of the array : 3
```

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

```
Enter array elements :
```

If the user gives the **input** as:

```
Enter array elements : 10 20 30
```

then the program should **print** the result as:

```
Array elements in reverse order : 30 20 10
```

[**Hint:** First read an integers from standard input into the array and then use a loop to iterate on that array in the reverse order (meaning starting from the last element till the first) to print the elements.]

Note: Do use the printf() function without a newline character (\n).

Source Code:

print.c

```
#include<stdio.h>
main()
{
   int k,a[100],n,b;
   printf("Enter size of the array : ");
   scanf("%d",&n);
   int size = a[n];
   printf("Enter array elements : " );
   for(k=0;k<n;k++)
   {
      scanf("%d",&a[k]);
   }
   printf("Array elements in reverse order : " );
   for(k=n-1;k>=0;k--)
   {
      printf("%d ",a[k]);
   }
   printf("\n");
   return 0;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter size of the array : 3
Enter array elements : 10 20 30
Array elements in reverse order : 30 20 10

Test Case - 2
User Output
Enter size of the array : 6
Enter array elements : 11 88 66 22 33 44
Array elements in reverse order : 44 33 22 66 88 11

S.No: 14	Exp. Name: ***Program - Addition of two matrices***	Date:2023-04-04

Aim:

The below sample code finds the **addition** of two matrices.

In the **main()** function read a two two-dimensional array of elements and then find the **addition** of two matrices.

The **logic** is
First checks the **row sizes** and **column sizes** of two two-dimensional arrays are equal or not.
If the sizes are not equal then print "Addition is not possible" and stop the process.
If the sizes are equal then use **two for loops** to add each corresponding elements of two matrices and finally print the result.
Fill in the missing code so that it produces the desired output.

Source Code:

matrix.c

```
#include<stdio.h>
void main()
{
   int i,j,m,n,p,q;
   int a[10][10],b[10][10],c[10][10];
   printf("Enter the row & column sizes of matrix-1 : ");
   scanf("%d %d",&m,&n);
   printf("Enter matrix-1 %d elements : ",m*n);
   for(i=0;i<m;i++)
   {
      for(j=0;j<n;j++)
      {
         scanf("%d",&a[i][j]);
      }
   }
   printf("Enter the row & column sizes of matrix-2 : ");
   scanf("%d %d",&p,&q);
   printf("Enter matrix-2 %d elements : ",p*q);
   for(i=0;i<p;i++)
   {
      for(j=0;j<q;j++)
      {
         scanf("%d",&b[i][j]);
      }
   }
   printf("The given matrix-1 is\n");
   for(i=0;i<m;i++)
   {
      for(j=0;j<n;j++)
      {
         printf("%d ",a[i][j]);
      }
      printf("\n");
   }
   printf("The given matrix-2 is\n");
   for(i=0;i<p;i++)
```

```
    {
        for(j=0;j<q;j++)
        {
            printf("%d ",b[i][j]);
        }
        printf("\n");
    }
    if(m==p&&n==q)
    {
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
            {
                c[i][j]=a[i][j]+b[i][j];
            }
        }
        printf("Addition of two matrices is\n");
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
            {
                printf("%d ",c[i][j]);
            }
            printf("\n");
        }
    }
    else
    {
        printf("Addition is not possible\n");
    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the row & column sizes of matrix-1 : 2 2
Enter matrix-1 4 elements : 1 2 3 4
Enter the row & column sizes of matrix-2 : 2 2
Enter matrix-2 4 elements : 4 5 6 7
The given matrix-1 is
1 2
3 4
The given matrix-2 is
4 5
6 7
Addition of two matrices is
5 7
9 11

ID: 224G1A0557
2022-2026-CSE-A
Srinivasa Ramanujan Institute of Technology

S.No: 15	Exp. Name: ***Program - Subtraction of two matrices***	Date:2023-04-04

Aim:

The below sample code finds the **subtraction** of two matrices.

In the **main()** function read a two two-dimensional array of elements and then find the **subtraction** of two matrices.

The **logic** is
First checks the **row sizes** and **column sizes** of two two-dimensional arrays are equal or not.
If the sizes are not equal then print "subtraction is not possible" and stop the process.
If the sizes are equal then use **two for loops** to subtract each corresponding elements of two matrices and finally print the result.
Fill in the missing code so that it produces the desired output.

Source Code:

submatrix.c

```
#include<stdio.h>
void main()
{
   int i,j,m,n,p,q;
   int a[10][10],b[10][10],c[10][10];
   printf("Enter the row & column sizes of matrix-1 : ");
   scanf("%d %d",&m,&n);
   printf("Enter matrix-1 %d elements : ",m*n);
   for(i=0;i<m;i++)
   {
      for(j=0;j<n;j++)
      {
         scanf("%d ",&a[i][j]);
      }
   }
   printf("Enter the row & column sizes of matrix-2 : \n");
   scanf("%d %d",&p,&q);
   printf("Enter matrix-2 %d elements : ",p*q);
   for(i=0;i<p;i++)
   {
      for(j=0;j<q;j++)
      {
      scanf("%d",&b[i][j]);
   }
}
printf("The given matrix-1 is\n");
for(i=0;i<m;i++)
{
   for(j=0;j<n;j++)
   {
      printf("%d ",a[i][j]);
   }
   printf("\n");
}
printf("The given matrix-2 is\n");
for(i=0;i<p;i++)
```

```
{
   for(j=0;j<q;j++)
   {
      printf("%d ",b[i][j]);
   }
   printf("\n");
}
if(m==p&&n==q)
{
   for(i=0;i<m;i++)
   {
      for(j=0;j<n;j++)
      {
         c[i][j]=a[i][j]-b[i][j];
      }
   }
   printf("Subtraction of two matrices is\n");
   for(i=0;i<m;i++)
   {
      for(j=0;j<n;j++)
      {
         printf("%d ",c[i][j]);
      }
      printf("\n");
   }
}
else
{
   printf("Subtraction is not possible\n");
}
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the row & column sizes of matrix-1 : 2 2
Enter matrix-1 4 elements : 6 4 8 1
Enter the row & column sizes of matrix-2 : 2 2
Enter matrix-2 4 elements : 1 2 3 4
The given matrix-1 is
6 4
8 1
The given matrix-2 is
1 2
3 4
Subtraction of two matrices is
5 2
5 -3

ID: 224G1A0557

2022-2026-CSE-A

Srinivasa Ramanujan Institute of Technology

S.No: 16	Exp. Name: ***Write a C program to perform matrix multiplication***	Date:2023-04-03

Aim:

Write a C program to perform matrix multiplication on two dimensional matrix.

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

```
Enter the row & column sizes of matrix-1 :
```

For example, if the user gives the input as:

```
Enter the row & column sizes of matrix-1 : 2 2
```

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

```
Enter matrix-1 4 elements :
```

If the user gives the input as:

```
Enter matrix-1 4 elements : 1 1 2 2
```

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

```
Enter the row & column sizes of matrix-2 :
```

If the user gives the input as:

```
Enter the row & column sizes of matrix-2 : 2 2
```

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

```
Enter matrix-2 4 elements :
```

If the user gives the input as:

```
Enter matrix-2 4 elements : 1 2 7 4
```

Then the program should print the result as:

```
The given matrix-1 is
1 1
2 2
The given matrix-2 is
1 2
7 4
Multiplication of two matrices is
8 6
16 12
```

Otherwise, the program should print the result as :

```
Multiplication is not possible
```

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

Source Code:

matmul.c

```
#include<stdio.h>
void main()
{
   int i,j,k,m,n,p,q;
   int a[5][5],b[5][5],c[5][5];
   printf("Enter the row & column sizes of matrix-1 : ");
   scanf("%d %d",&m,&n);
   printf("Enter matrix-1 %d elements : ", m*n);
   for(i=0;i<m;i++)
   {
      for(j=0;j<n;j++)
      {
         scanf("%d",&a[i][j]);
      }
   }
   printf("Enter the row & column sizes of matrix-2 : ");
   scanf("%d %d",&p,&q);
   printf("Enter matrix-2 %d elements : ", p*q);
   for(i=0;i<p;i++)
   {
      for(j=0;j<q;j++)
      {
         scanf("%d",&b[i][j]);
      }
   }
   printf("The given matrix-1 is\n");
   for(i=0;i<m;i++)
   {
      for(j=0;j<n;j++)
      {
         printf("%d ",a[i][j]);
      }
      printf("\n");
   }
   printf("The given matrix-2 is\n");
   for(i=0;i<p;i++)
   {
      for(j=0;j<q;j++)
      {
         printf("%d ",b[i][j]);
      }
      printf("\n");
   }
      if(n==p)
      {
```


ID: 224G1A0557

```
        for (i=0;i<m;i++)
        {
           for(j=0;j<q;j++)
           {
              c[i][j]=0 ;
              for(k=0;k<p;k++)
              {
                 c[i][j]=c[i][j]+a[i][k]*b[k][j] ;
              }
           }
        }
        printf("Multiplication of two matrices is\n");
        for(i=0;i<m;i++)
        {
           for(j=0;j<q;j++)
           {
              printf("%d ",c[i][j]);
           }
           printf("\n");
        }
     }
     else
     {
        printf("Multiplication is not possible\n");
     }
  }
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the row & column sizes of matrix-1 : 2 2
Enter matrix-1 4 elements : 1 2 3 4
Enter the row & column sizes of matrix-2 : 2 2
Enter matrix-2 4 elements : 4 5 6 7
The given matrix-1 is
1 2
3 4
The given matrix-2 is
4 5
6 7
Multiplication of two matrices is
16 19
36 43

Test Case - 2
User Output
Enter the row & column sizes of matrix-1 : 2 2
Enter matrix-1 4 elements : 1 1 2 2
Enter the row & column sizes of matrix-2 : 2 2
Enter matrix-2 4 elements : 1 2 7 4
The given matrix-1 is
1 1
2 2
The given matrix-2 is
1 2
7 4
Multiplication of two matrices is
8 6
16 12

ID: 224G1A0557
2022-2026-CSE-A
Srinivasa Ramanujan Institute of Technology

S.No: 17	Exp. Name: ***Write a C program to implement the string manipulation operations by using library functions.***	Date:2023-04-02

ID: 224G1A0557

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");
}
```

2022-2026-CSE-A

Srinivasa Ramanujan Institute of Technology

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

ID: 224G1A0557
2022-2026-CSE-A
Srinivasa Ramanujan Institute of Technology

S.No: 18	Exp. Name: ***given a list of n numbers, Design an algorithm which prints the number of stars equivalent to the value of the number. The stars for each number should be printed horizontally.***	Date:2023-04-01

ID: 224G1A0557

2022-2026-CSE-A

Srinivasa Ramanujan Institute of Technology

Aim:

Take a list of n numbers, Design an algorithm which prints the number of stars equivalent to the value of the number. The stars for each number should be printed horizontally.

Sample input output

```
Sample input output -1:

Enter the number of numbers: 6
Enter number 1: 4
Enter number 2: 6
Enter number 3: 9
Enter number 4: 5
Enter number 5: 2
Enter number 6: 6
****
******
*********
*****
**
******

Sample input output -2:

Enter the number of numbers: 4
Enter number 1: 4
Enter number 2: 2
Enter number 3: 1
Enter number 4: 3
****
**
*
***
```

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

Source Code:

star.c

```
#include<stdio.h>
main()
{
   int n,j,i,a[10];
   printf("Enter the number of numbers: ");
   scanf("%d",&n);
   for(i=0;i<n;i++)
   {
      printf("Enter number %d: ",i+1);
      scanf("%d",&a[i]);
   }
```

```
   for(i=0;i<n;i++)
   {
      for(j=1;j<=a[i];j++)
      {
         printf("*");
      }
      printf("\n");
   }
}
```


ID: 224G1A0557

Srinivasa Ramanujan Institute of Technology

2022-2026-CSE-A

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the number of numbers: 6
Enter number 1: 4
Enter number 2: 6
Enter number 3: 9
Enter number 4: 5
Enter number 5: 2
Enter number 6: 6
****
******
****
*****
**
******

Test Case - 2
User Output
Enter the number of numbers: 5
Enter number 1: 5
Enter number 2: 4
Enter number 3: 3
Enter number 4: 2
Enter number 5: 1
*****
****
***
**
*

ID: 224G1A0557
2022-2026-CSE-A
Srinivasa Ramanujan Institute of Technology

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

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

```
#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");
}
```


ID: 224G1A0557

Srinivasa Ramanujan Institute of Technology

2022-2026-CSE-A

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter n value : 3
Enter 3 elements : 45 67 34
Elements before sorting : 45 67 34
Elements after sorting : 34 45 67

ID: 224G1A0557

2022-2026-CSE-A

Srinivasa Ramanujan Institute of Technology

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

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

```
#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;
         }
      }
   }
}
```


ID: 224G1A0557

sorta.c

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


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