

S.No: 1	Exp. Name: <i>Write a C program to find the reverse of a given number</i>	Date: 2024-01-30
---------	---	------------------

Aim:

Design a C program which reverses the given number.

Source Code:

reverse.c

```
#include<stdio.h>
int main()
{
    int n,rem=0,rev=0;
    scanf("%d",&n);
    while(n>0)
    {
        rem=n%10;
        rev=rev*10+rem;
        n=n/10;
    }
    printf("Reversed number= %d",rev);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
456
Reversed number= 654

Test Case - 2
User Output
958745
Reversed number= 547859

S.No: 2	Exp. Name: <i>Write a C program to find second largest for the given numbers</i>	Date: 2024-01-30
---------	--	------------------

Aim:

Design a C program which finds the second maximum number among the given one dimensional array of elements.

Sample Input and Output: Enter how many values you want to read : 6
Enter the value of a[0] : 45
Enter the value of a[1] : 24
Enter the value of a[2] : 23
Enter the value of a[3] : 65
Enter the value of a[4] : 78
Enter the value of a[5] : 42
The second largest element of the array = 65

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

Source Code:

second_large.c

```

#include<stdio.h>

void main()
{
    int i,n,a[20],max1=0,max2=0;

    printf("Enter how many values you want to read : ");

    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        printf("Enter the value of a[%d] : ",i);

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

    }

    for(i=0;i<n;i++)
    {
        if(max1<a[i])
        {
            max2=max1;

            max1=a[i];

        }

        else if(a[i]>max2&&a[i]<max1)
        {
            max2=a[i];

        }

    }

    printf("The second largest element of the array = %d\n",max2);

}

```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output
Enter how many values you want to read :
4
Enter the value of a[0] :
32
Enter the value of a[1] :
25
Enter the value of a[2] :
69
Enter the value of a[3] :
47
The second largest element of the array = 47

S.No: 3	Exp. Name: <i>Write a program which finds the kth smallest number among the given list of numbers.</i>	Date: 2024-01-30
---------	--	------------------

Aim:

Write a program which finds the k^{th} smallest number among the given one dimensional array.

Sample Input and Output:

```
Enter how many values you want to read : 5
Enter the value of a[0] : 20
Enter the value of a[1] : 30
Enter the value of a[2] : 16
Enter the value of a[3] : 15
Enter the value of a[4] : 1
Enter which smallest element you want: 2
16 is the 2th smallest element
```

Hint: The k^{th} element refers to the index.

Source Code:

```
smallest.c
```

```

#include<stdio.h>

void main()
{
    int kth,i,j,n,a[20],temp,pos;

    printf("Enter how many values you want to read : ");

    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        printf("Enter the value of a[%d] : ",i);

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

    printf("Enter which smallest element you want: " );

    scanf("%d",&kth);

    for(i=0;i<n;i++)
    {
        pos=i;

        for(j=i+1;j<n;j++)
            if(a[j]<a[pos])
            {
                pos=j;
            }

        temp=a[i];
        a[i]=a[pos];
        a[pos]=temp;
    }

    printf("%d is the %dth smallest element",a[kth],kth);
}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter how many values you want to read :
5
Enter the value of a[0] :
20
Enter the value of a[1] :
30
Enter the value of a[2] :
16
Enter the value of a[3] :
15
Enter the value of a[4] :
1
Enter which smallest element you want:
2
16 is the 2th smallest element

Page No: 7

ID: 224G1A0546

2022-2026-CSE-A

Srinivasa Ramanujan Institute of Technology

Test Case - 2
User Output
Enter how many values you want to read :
6
Enter the value of a[0] :
32
Enter the value of a[1] :
65
Enter the value of a[2] :
98
Enter the value of a[3] :
74
Enter the value of a[4] :
12
Enter the value of a[5] :
15
Enter which smallest element you want:
4
74 is the 4th smallest element

b = 3
c = 4
d = 1

Test Case - 2
User Output
Enter values of a, b, c and d:
98 74 21 36
After swapping
a = 74
b = 21
c = 36
d = 98

S.No: 5	Exp. Name: <i>Write a program to find the count of positive and negative numbers</i>	Date: 2024-01-30
---------	--	------------------

Aim:

Develop a C Program which counts the number of positive and negative numbers separately and also compute the sum of them.

```
Sample Input and Output:  
How many numbers you want to add : 6  
Enter number a[0] : 3  
Enter number a[1] : 5  
Enter number a[2] : -5  
Enter number a[3] : 7  
Enter number a[4] : -8  
Enter number a[5] : 6  
Count of positive numbers = 4  
Sum of positive numbers = 21  
Count of negative numbers = 2  
Sum of Negative numbers = -13
```

Source Code:

```
count.c
```

```

#include<stdio.h>

int main()
{
    int a[20],n,i,sump=0,sumn=0,countp=0,countn=0;

    printf("How many numbers you want to add : ");

    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        printf("Enter number a[%d] : ",i);

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

    for(i=0;i<n;i++)
    {
        if(a[i]>0)
        {
            sump += a[i];

            countp = countp + 1;
        }
        else
        {
            sumn += a[i];

            countn = countn + 1;
        }
    }

    printf("Count of positive numbers = %d\n",countp);

    printf("Sum of positive numbers = %d\n",sump);

    printf("Count of negative numbers = %d\n",countn);

    printf("Sum of Negative numbers = %d\n",sumn);
}

```

```
}
```

Execution Results - All test cases have succeeded!

Page No. 12

ID: 224G1A0546

2022-2026-CSE-A

Srinivasa Ramanujan Institute of Technology

Test Case - 1
User Output
How many numbers you want to add :
5
Enter number a[0] :
4
Enter number a[1] :
5
Enter number a[2] :
6
Enter number a[3] :
2
Enter number a[4] :
6
Count of positive numbers = 5
Sum of positive numbers = 23
Count of negative numbers = 0
Sum of Negative numbers = 0

Test Case - 2
User Output
How many numbers you want to add :
4
Enter number a[0] :
-4
Enter number a[1] :
-1
Enter number a[2] :
-3
Enter number a[3] :
-2
Count of positive numbers = 0
Sum of positive numbers = 0
Count of negative numbers = 4
Sum of Negative numbers = -10

S.No: 6	Exp. Name: <i>Implement the C program which computes the sum of the first n terms of the series</i>	Date: 2024-01-30
---------	---	------------------

Aim:

Implement the C program which computes the sum of the first n terms of the series

Sum = 1 - 3 + 5 - 7 + 9 +

Sample Input and Output - 1:

Enter the value of n: 99
The sum of first 99 terms of the series is: 99

Source Code:

sum.c

```
#include<stdio.h>

int main()
{
    int n, i, sum=0,sumn=0,sump=0;

    printf("Enter the value of n: ");

    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        if(i%2==0)
        {
            sump += 2*i+1;
        }
        else
        {
            sumn += -(2*i+1);
        }
    }

    sum=sump + sumn;

    printf("The sum of first %d terms of the series is: %d\n",n,sum);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the value of n:
789
The sum of first 789 terms of the series is: 789

Test Case - 2
User Output
Enter the value of n:
76
The sum of first 76 terms of the series is: -76

Test Case - 3
User Output
Enter the value of n:
99
The sum of first 99 terms of the series is: 99

S.No: 7	Exp. Name: <i>Design a C program which determines factorial of numbers</i>	Date: 2024-01-30
---------	--	------------------

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>

int 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(max>=fact&&fact>=min)
        {
            printf("%d ",i);
        }
    }

}
```

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: 8	Exp. Name: <i>Design an algorithm and implement using a C program which finds the sum of the infinite series</i>	Date: 2024-01-30
---------	--	------------------

Aim:

Design an algorithm and implement using a C program which finds the `sum` of the `infinite series`

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots,$$

Print the result as shown in the example.

Sample Input and Output:

Enter the value of x and n: 4 5
sum = 3.666667

Source Code:

infinite.c

```

#include<stdio.h>

#include<math.h>

int main()
{
    int x,n,m,i=0,fact=1;

    float k,sum=0;

    printf("Enter the value of x and n: ");

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

    while(i<=n)
    {
        if(i%2==0)
        {
            fact=1;

            for(m=1;m<=i;m++)
            {
                fact=fact*m;
            }

            k=(pow(x,i))/fact;
        }

        if(i%4!=0)
        {
            fact=1;

            for(m=1;m<=i;m++)
            {
                fact=fact*m;
            }

            k=-(pow(x,i))/fact;
        }
    }
}

```

```

        i=i+2;

    }

    printf("sum = %f",sum);

}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the value of x and n:
4 5
sum = 3.666667

Test Case - 2
User Output
Enter the value of x and n:
12 5
sum = 793.000000

S.No: 9	Exp. Name: <i>Design a C program to print the sequence of numbers in which each number is the sum of the three most recent predecessors</i>	Date: 2024-01-30
---------	---	------------------

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 a=0,b=1,c=1,d,temp,n,i;

    printf("Enter the number of terms: ");

    scanf("%d",&n);

    printf("First %d terms in the series are:\n",n);

    printf("%d\n%d\n%d\n",a,b,c);

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

        printf("%d\n",d);

        temp=a;

        a=b;

        b=c;

        c=d;
    }

}

```

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: 10	Exp. Name: <i>Write a C program to convert a Decimal number into binary, octal and hexadecimal number using a single user defined function.</i>	Date: 2024-01-30
----------	---	------------------

Aim:

Write a C program to convert a Decimal number into binary, octal and hexadecimal number using a single user defined function.

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

Enter a positive decimal number :

For example, if the user gives the input as:

Enter a positive decimal number : 789

then the program should print the result as:

The binary number of decimal 789 is : 1100010101

The octal number of decimal 789 is : 1425

The hexadecimal number of decimal 789 is : 315

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

Source Code:

oche.c

```

#include<stdio.h>

int main()
{
    int n,s,temp,bin[100],i,j;

    printf("Enter a positive decimal number : ");

    scanf("%d",&n);

    s=2*n;

    s=s/2;

    temp=s;

    for(i=0;s>0;i++)
    {
        bin[i]=s%2;

        s=s/2;
    }

    printf("The binary number of decimal %d is : ",temp);

    for(j=i-1;j>=0;j--)

        printf("%d",bin[j]);

    printf("\n");

    printf("The octal number of decimal %d is : %o\n",n,n);

    printf("The hexadecimal number of decimal %d is : %X\n",n,n);
}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter a positive decimal number :
45
The binary number of decimal 45 is : 101101
The octal number of decimal 45 is : 55
The hexadecimal number of decimal 45 is : 2D

Test Case - 2
User Output
Enter a positive decimal number :
10
The binary number of decimal 10 is : 1010
The octal number of decimal 10 is : 12
The hexadecimal number of decimal 10 is : A

Test Case - 3
User Output
Enter a positive decimal number :
6789
The binary number of decimal 6789 is : 1101010000101
The octal number of decimal 6789 is : 15205
The hexadecimal number of decimal 6789 is : 1A85

S.No: 11	Exp. Name: <i>Develop an algorithm which computes the all the factors between 1 to 100 for a given number and implement it using C.</i>	Date: 2024-01-30
----------	---	------------------

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   39   78
```

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>

int 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: 13591545						

S.No: 13	Exp. Name: <i>Write a C program to display the elements of an array in reverse order</i>	Date: 2024-01-30
----------	--	------------------

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>

int main()
{
    int a[20],i,n;

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

    scanf("%d",&n);

    printf("Enter array elements : ");

    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }

    printf("Array elements in reverse order : ");

    for(i=n-1;i>=0;i--)
    {
        printf("%d ",a[i]);
    }

    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

6
Enter array elements :
11 88 66 22 33 44
Array elements in reverse order : 44 33 22 66 88 11


```

#include<stdio.h>

void main()
{
    int a[5][5],b[5][5],c[5][5];

    int i,j,k,m,n,p,q;

    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++)
    {

```



```

        {
            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)

    {

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

        {

            for(j=0;j<q;j++)

            {

                c[i][j]=0;

                for(k=0;k<n;k++)

                {

                    c[i][j]=c[i][j]+a[i]

[k]*b[k][j];

                }

            }

        }

    }

```

```

        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

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

S.No: 17	Exp. Name: <i>Write a C program to implement the string manipulation operations by using library functions.</i>	Date: 2024-01-30
----------	---	------------------

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>

int main()
{
    char str1[100],str2[100];

    int l;

    printf("Enter two strings : ");

    scanf("%s%s",str1,str2);

    l=strlen(str1);

    printf("The length of %s : %d\n",str1,l);

    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");
    }

    else if(i>0)
    {
        printf("%s is greater than %s\n",str1,str2);
    }

    else
    {
        printf("%s is less thsn %s\n",str1,str2);
    }

    printf("The concatenated string : %s\n",strcat(str1,str2));
}

```

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: 18	Exp. Name: <i>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.</i>	Date: 2024-01-30
----------	--	------------------

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>

int main()
{
    int a[20];

    int i,j,p;

    printf("Enter the number of numbers: ");

    scanf("%d",&j);

    for(i=0;i<j;i++)
    {
        printf("Enter number %d: ",i+1);

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

    }

    for(i=0;i<j;i++)
    {
        for(p=1;p<=a[i];p++)
        {
            printf("*");

        }

        printf("\n");

    }

}

```

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	

**	
*	


```

#include<stdio.h>

int 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=1;

        for(j=i;j>=0;j--)
        {

            if(a[j]>=a[max])

                max=j;

        }

        temp=a[i];

        a[i]=a[max];

        a[max]=temp;

    }
}

```

```

    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

S.No: 23	Exp. Name: <i>Illustrate the use of auto variable</i>	Date: 2024-01-30
----------	---	------------------

Aim:

Illustrate the use of auto variable.

The variables defined using **auto** storage class are called as local variables.

Auto stands for **automatic** storage class. A variable is in auto storage class by default if it is not explicitly specified.

The scope of an auto variable is **limited with the particular block only**.

Once the control goes out of the block, the access is destroyed. This means only the block in which the auto variable is declared can access it.

A keyword **auto** is used to define an auto storage class. By default, an auto variable contains a **garbage value**.

Follow the instructions given in the comment lines to declare auto variables and print their values at different places in the program.

Source Code:

auto.c

```

#include<stdio.h>

void main() {

    // Declare an auto variable d of type integer.

    // Print the value of d.

    {

        auto int d=32767;

        printf("%d\n",d);

        // Declare and initialize the auto variable d with 4.

        {

            auto int d=4;

            {

                auto int d=6;

                printf("%d\n",d);

            }

            printf("%d\n",d);

            // Declare and initialize the auto variable d with 6/

            // Print the value of d.

        }

        // Print the value of d.

    }

}

```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
32767
6
4

S.No: 27	Exp. Name: <i>Develop a C program which takes two numbers as command line arguments and finds all the common factors of those two numbers.</i>	Date: 2024-01-30
----------	--	------------------

Aim:

Develop a C program which takes two numbers as command line arguments and finds all the common factors of those two numbers.

Sample input output

```
Sample input output -1:
Cmd Args : 10 20
Common factors for 10 and 20 are: 1 2 5 10
Sample input output -2:
Cmd Args : 45 23
Common factors for 45 and 23 are: 1
```

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

Source Code:

common_factors.c

```
#include<stdio.h>

#include<stdlib.h>

int main(int argc,char*argv[])
{
    int a,b;

    int i,small;

    a=atoi(argv[1]);

    b=atoi(argv[2]);

    small=(a<b)?a:b;

    printf("Common factors for %d and %d are: ",a,b);

    for(i=1;i<=small;i++)
    {
        if(a%i==0&&b%i==0)

            printf("%d\t",i);

    }

    printf("\n");

}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Common factors for 10 and 20 are: 1 2 5 10

Test Case - 2
User Output
Common factors for 18 and 39 are: 1 3

