

| | | |
|---------|--|------------------|
| S.No: 1 | Exp. Name: <i>Write a C program to find the reverse of a given number</i> | Date: 2023-04-18 |
|---------|--|------------------|

Aim:

Design a C program which reverses the given number.

Source Code:

reverse.c

```
#include<stdio.h>
void 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: **Write a C program to find second largest for the given numbers**

Date: 2023-04-18

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>
int 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: **Write a program which finds the kth smallest number among the given list of numbers.**

Date: 2023-04-18

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>
#define MAX 100
int main()
{
    int a[MAX],i,n,j,kth,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

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

S.No: 4

Exp. Name: **Design an algorithm and implement using C language the following exchanges**

Date: 2023-04-18

Aim:

Design an algorithm and implement using C language the following exchanges $a \leftarrow b \leftarrow c \leftarrow d \leftarrow a$ and print the result as shown in the example.

Sample Input and Output:
Enter values of a, b, c and d: 98 74 21 36
After swapping
a = 74
b = 21
c = 36
d = 98

Source Code:

exchange.c

```
#include<stdio.h>
void main()
{
    int a,b,c,d,temp;
    printf("Enter values of a, b, c and d: ");
    scanf("%d%d%d%d",&a,&b,&c,&d);
    temp=a;
    a=b;
    b=c;
    c=d;
    d=temp;
    printf("After swapping\na = %d\nb = %d\nc = %d\n d = %d\n",a,b,c,d);
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

Enter values of a, b, c and d:

1 2 3 4

After swapping

a = 2

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: **Write a program to find the count of positive and negative numbers**

Date: 2023-04-18

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!

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