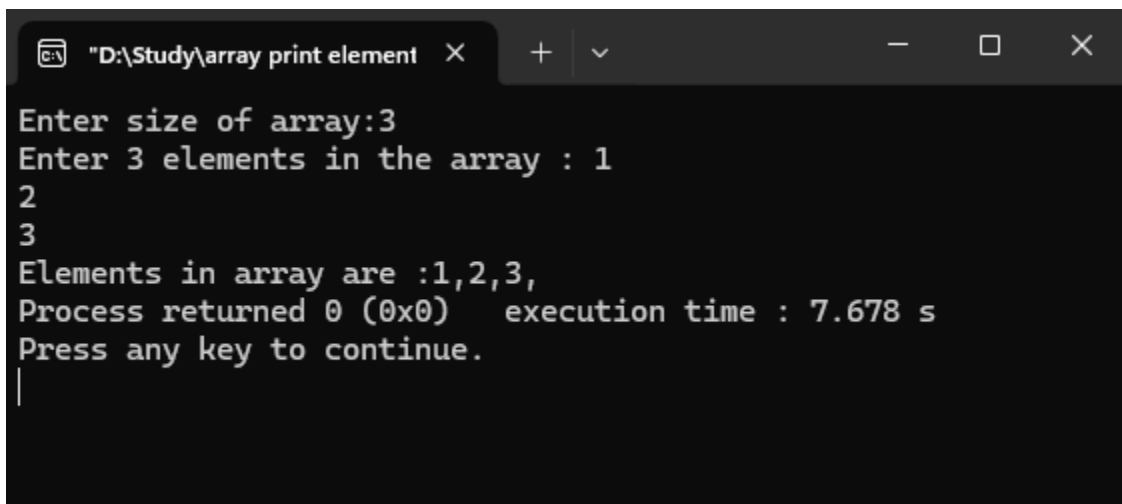


**Problem-1:** Write a C program to read and print elements of array. – using recursion.

**Source Code:**

```
#include<stdio.h>
int main()
{
    int n;int i;
    printf("Enter size of array:");
    scanf("%d",&n);
    int arr[n];
    printf("Enter %d elements in the array : ",n);
    for (i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    printf("Elements in array are :");
    for (i=0;i<n;i++)
    {
        printf("%d,",arr[i]);
    }
    return 0;
}
```

**Output:**

A screenshot of a Windows command prompt window titled "D:\Study\array print element". The window shows the execution of a C program. The user enters "3" for the size of the array. Then, the user enters "1", "2", and "3" as the elements of the array. The program outputs "Elements in array are :1,2,3,". Below this, it shows "Process returned 0 (0x0) execution time : 7.678 s" and "Press any key to continue." with a cursor on a new line.

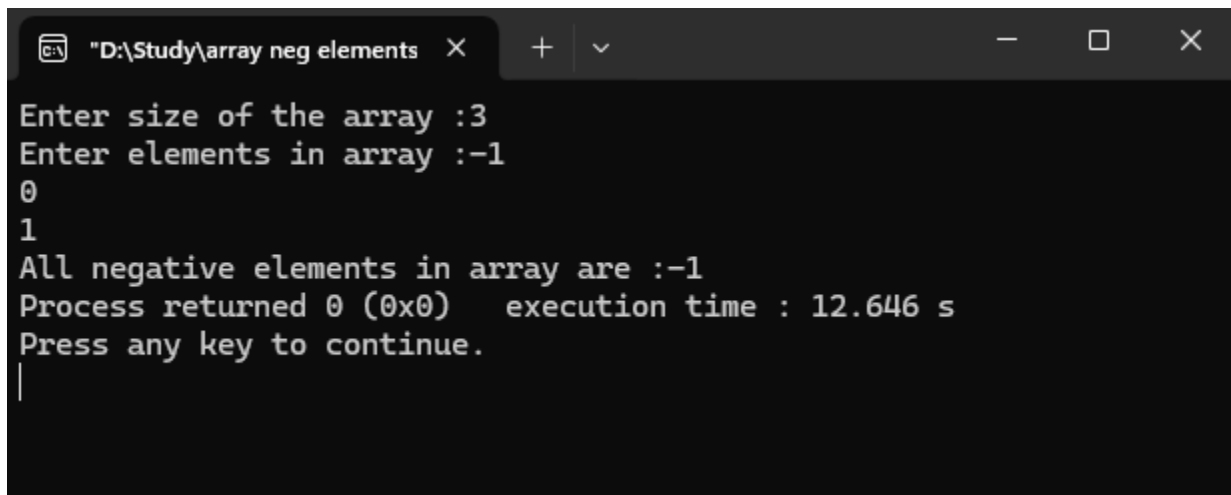
```
"D:\Study\array print element" X + - □ X
Enter size of array:3
Enter 3 elements in the array : 1
2
3
Elements in array are :1,2,3,
Process returned 0 (0x0) execution time : 7.678 s
Press any key to continue.
|
```

**Problem-2:** Write a C program to print all negative elements in an array.

**Source Code:**

```
#include<stdio.h>
int main(){
    int i,n ;
    printf("Enter size of the array :");
    scanf("%d",&n);
    int arr[n];
    printf("Enter elements in array :");
    for (i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    printf("All negative elements in array are :");
    for(i=0;i<n;i++)
    {
        if(arr[i]<0)
            printf("%d ",arr[i]);
    }
    return 0;
}
```

**Output:**



```
"D:\Study\array neg elements" X + v - □ X
Enter size of the array :3
Enter elements in array :-1
0
1
All negative elements in array are :-1
Process returned 0 (0x0)   execution time : 12.646 s
Press any key to continue.
|
```

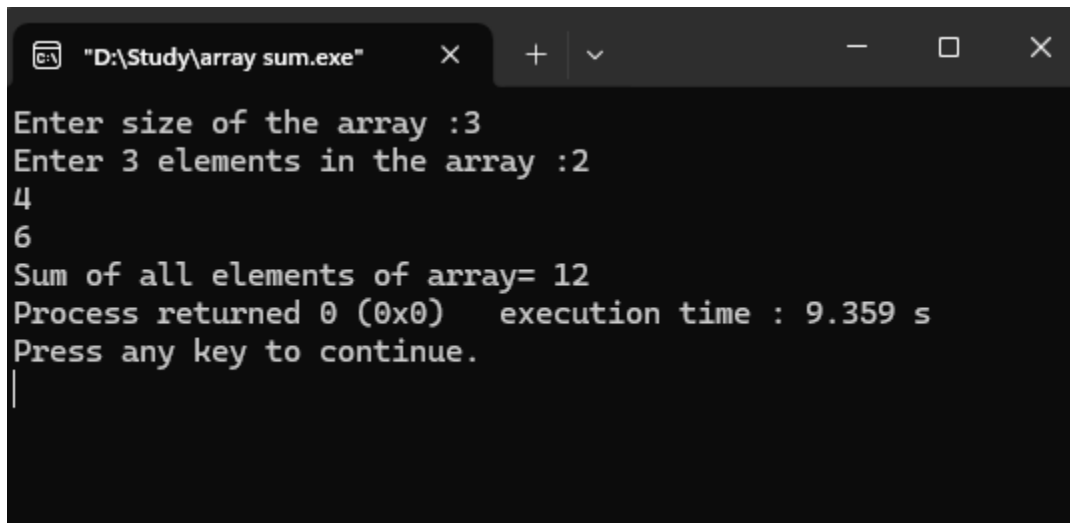
**Problem-3:** Write a C program to find sum of all array elements. – using recursion.

**Source Code:**

```
#include <stdio.h>

int main()
{
    int i,n,sum=0 ;
    printf("Enter size of the array :");
    scanf("%d",&n);
    int arr[n];
    printf("Enter %d elements in the array :",n);
    for (i=0;i<n;i++)
    {
        scanf("%d",&arr[i]);
    }
    printf("Sum of all elements of array= ");
    for(i=0;i<n;i++)
    {
        sum+=arr[i];
    }
    printf("%d ",sum);
    return 0;
}
```

**Output:**



```
"D:\Study\array sum.exe" × + ▾ − □ ×

Enter size of the array :3
Enter 3 elements in the array :2
4
6
Sum of all elements of array= 12
Process returned 0 (0x0)   execution time : 9.359 s
Press any key to continue.
|
```

**Problem-4:** [Write a C program to find maximum and minimum element in an array.](#) – using [recursion](#).

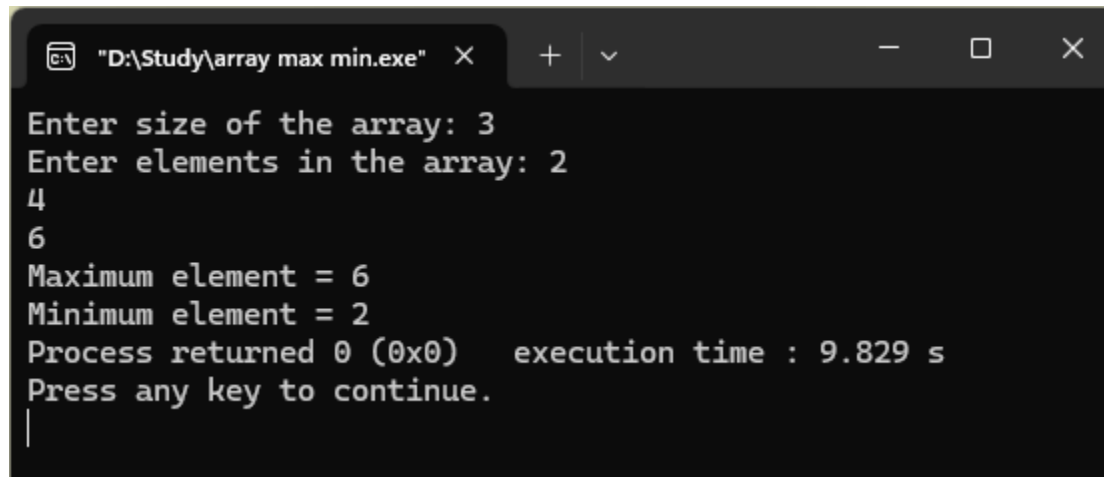
**Source Code:**

```
#include <stdio.h>
int main()
{
    int size, i, max, min;
    printf("Enter size of the array: ");
    scanf("%d", &size);
    int arr[size];
    printf("Enter elements in the array: ");
    for(i = 0; i < size; i++)
    {
        scanf("%d", &arr[i]);
    }
    max = arr[0];
    min = arr[0];

    for(i = 1; i < size; i++)
    {
        if(arr[i] > max)
        {
            max = arr[i];
        }
        if(arr[i] < min)
        {
            min = arr[i];
        }
    }

    printf("Maximum element = %d\n", max);
    printf("Minimum element = %d", min);
    return 0;
}
```

**Output:**



```
"D:\Study\array max min.exe" X + v - □ X
Enter size of the array: 3
Enter elements in the array: 2
4
6
Maximum element = 6
Minimum element = 2
Process returned 0 (0x0)   execution time : 9.829 s
Press any key to continue.
|
```

Problem number 5:

```
#include <stdio.h>

int main() {
    int flag = 0, position, goru[50] = {5, 1, 0, -15, 10, 3, 7, 100}, i, search_value;
    printf("Enter search_value: ");
    scanf("%d", &search_value);
    for (i = 0; i < 8; i++) {
        if (search_value == goru[i]) {
            flag = 1;
            position = i;
            break;
        }
    }
    if (flag == 1)
        printf("%d is found and position = %d\n", search_value, position + 1);
    else
        printf("Value is not found\n");
    return 0;
}
```

OUTPUT:

```
Enter search_value: 5
5 is found and position = 1
```