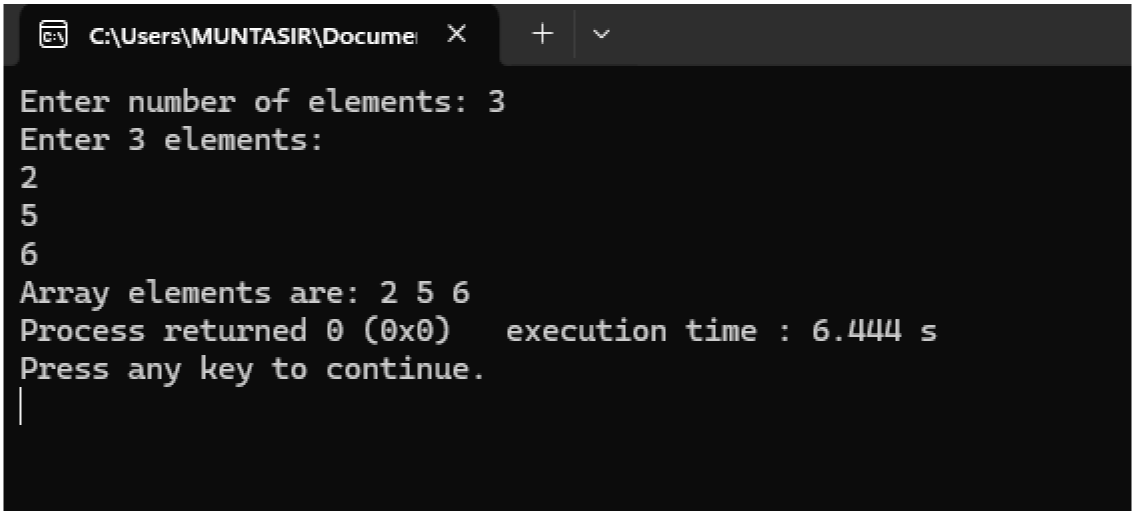
Problem Name : Write a C program to read and print elements of array. – using recursion.

Source Code:

```
#include <stdio.h>
void print(int a[], int i, int n) {
   if(i == n) return;
   printf("%d ", a[i]);
   print(a, i+1, n);
}
int main() {
   int a[50], n, i;
   printf("Enter size: ");
   scanf("%d", &n);
   printf("Enter elements: ");
   for(i=0; i<n; i++)
      scanf("%d\n", &a[i]);
   printf("Array: ");
   print(a, 0, n);
   return 0;
}
```

Output :



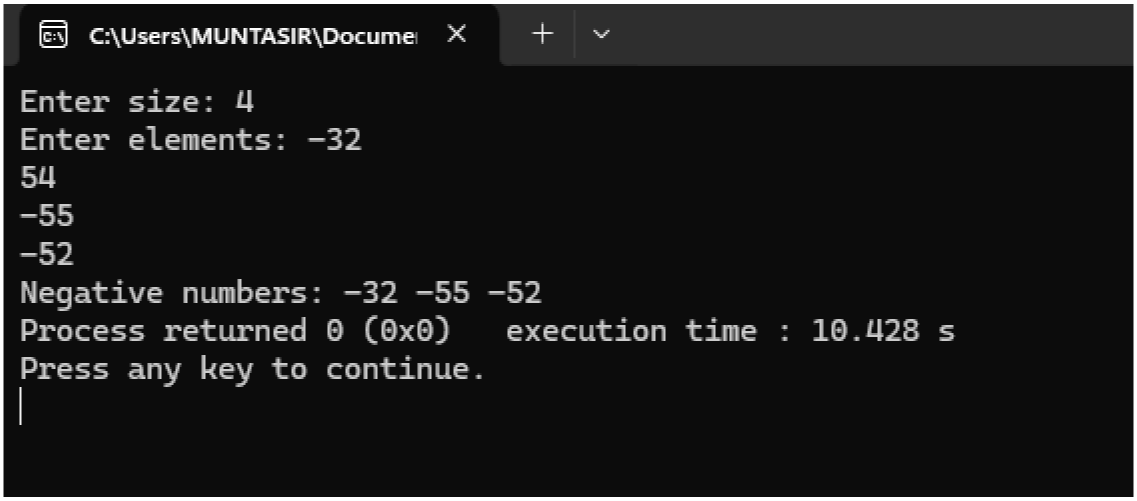
```
Enter number of elements: 3
Enter 3 elements:
2
5
6
Array elements are: 2 5 6
Process returned 0 (0x0)   execution time : 6.444 s
Press any key to continue.
```

Problem Name : Write a C program to print all negative elements in an array.

Source Code:

```
#include <stdio.h>
int main() {
   int a[50], n, i;
   printf("Enter size: ");
   scanf("%d", &n);
   printf("Enter elements: ");
   for(i=0; i<n; i++)
      scanf("%d", &a[i]);
   printf("Negative numbers: ");
   for(i=0; i<n; i++)
      if(a[i] < 0)
         printf("%d ", a[i]);
   return 0;
}
```

Output :



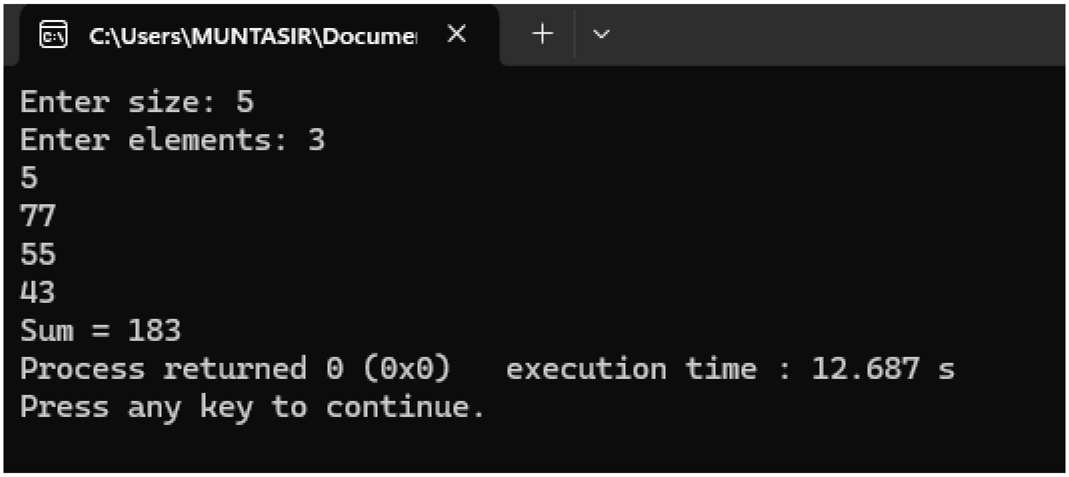
```
Enter size: 4
Enter elements: -32
54
-55
-52
Negative numbers: -32 -55 -52
Process returned 0 (0x0)   execution time : 10.428 s
Press any key to continue.
```

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

Source Code:

```
#include <stdio.h>
int sum(int a[], int n) {
   if(n == 0) return 0;
   return a[n-1] + sum(a, n-1);
}
int main() {
   int a[50], n, i;
   printf("Enter size: ");
   scanf("%d", &n);
   printf("Enter elements: ");
   for(i=0; i<n; i++)
      scanf("%d", &a[i]);
   printf("Sum = %d", sum(a, n));
   return 0;
}
```

Output :



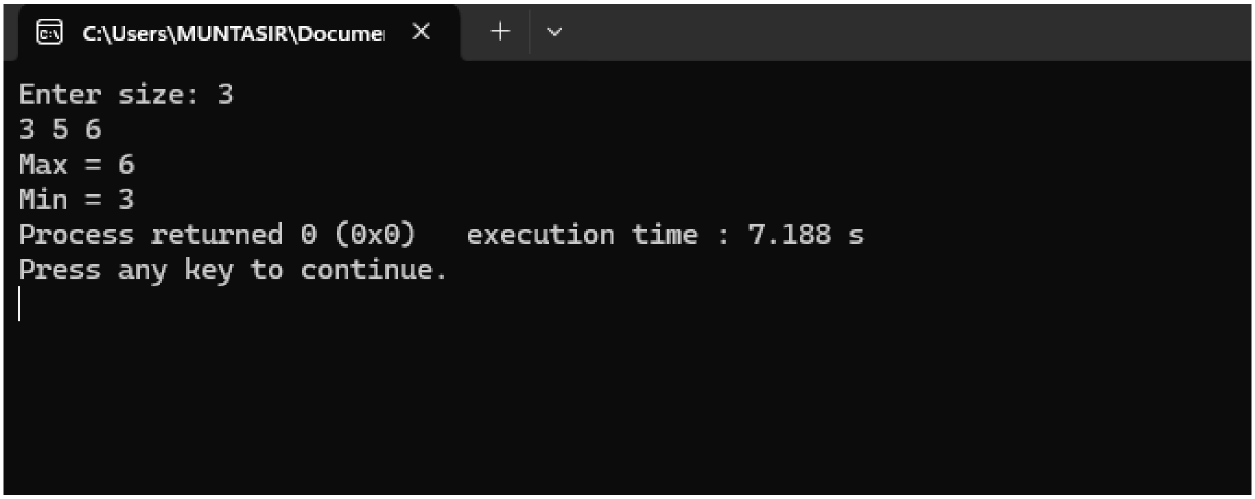
```
Enter size: 5
Enter elements: 3
5
77
55
43
Sum = 183
Process returned 0 (0x0)   execution time : 12.687 s
Press any key to continue.
```

Problem Name : Write a C program to find maximum and minimum element in an array.

Source Code:

```
#include <stdio.h>
int max(int a[], int n) {
  if(n == 1) return a[0];
  int m = max(a, n-1);
  return (a[n-1] > m) ? a[n-1] : m; }
int min(int a[], int n) {
  if(n == 1) return a[0];
  int m = min(a, n-1);
  return (a[n-1] < m) ? a[n-1] : m; }
int main() {
  int a[50], n, i;
  printf("Enter size: ");
  scanf("%d", &n);
  for(i=0; i<n; i++)
    scanf("%d", &a[i]);
  printf("Max = %d\nMin = %d", max(a,n), min(a,n));
  return 0; }
```

Output :



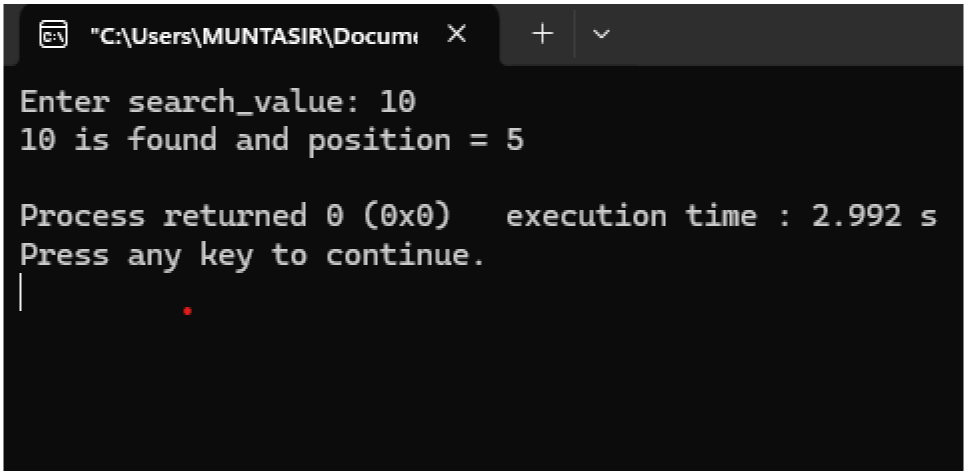
```
Enter size: 3
3 5 6
Max = 6
Min = 3
Process returned 0 (0x0)   execution time : 7.188 s
Press any key to continue.
```

Problem Name : Write a C program to search for a value in an array using linear search.

Source Code:

```
#include <stdio.h>
int main() {
    int flag = 0, position;
    int goru[50] = {5, 1, 0, -15, 10, 3, 7, 100};
    int 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: 10
10 is found and position = 5

Process returned 0 (0x0)   execution time : 2.992 s
Press any key to continue.
```