

Array

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

Input:

```
#include<stdio.h>

int main(){

    int i,N;

    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("\nElements in array are: ");

    for(i=0;i<N;i++){

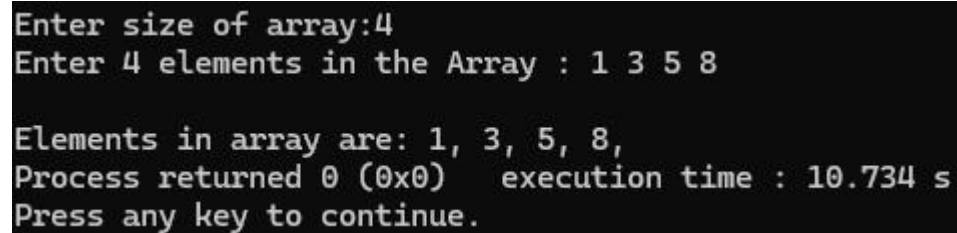
        printf("%d, ",arr[i]);

    }

    return 0;

}
```

Output:

A screenshot of a terminal window showing the output of the C program. The text is as follows:

```
Enter size of array:4
Enter 4 elements in the Array : 1 3 5 8

Elements in array are: 1, 3, 5, 8,
Process returned 0 (0x0)   execution time : 10.734 s
Press any key to continue.
```

2. Write a C program to print all negative elements in an array.

Input:

```
#include<stdio.h>

int main(){
```

```

int i,N;

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("\nAll negative elements in array are: ");

for(i=0;i<N;i++){

    if(arr[i]<0){

        printf("%d, ",arr[i]);

    }

}

return 0;

}

```

Output:

```

Enter size of the array:4
Enter 4 elements in the Array : 2 -1 -4 6

All negative elements in array are: -1, -4,
Process returned 0 (0x0)   execution time : 36.525 s
Press any key to continue.

```

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

Input:

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

```

int main(){

    int i,N,sum=0;

    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]);

}

for(i=0;i<N;i++){

    sum=sum + arr[i];

}

printf("Sum of all elements of array = %d", sum);

return 0;

}

```

Output:

```

Enter size of array:5
Enter 5 elements in the Array : 1 2 4 6 8
Sum of all elements of array = 21
Process returned 0 (0x0)   execution time : 33.737 s
Press any key to continue.

```

4. Write a C program to find maximum and minimum element in an array.– using recursion.

Input:

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

```
#define MAX_SIZE 100
```

```
int main(){
```

```
    int arr[MAX_SIZE];
```

```
    int i, max, min, size;
```

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

```
    scanf("%d", &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:

```

Enter size of the array: 10
Enter elements in the array: 10 -10 12 -1 0 25 -23 22 7 -9
Maximum element = 25
Minimum element = -23
Process returned 0 (0x0)   execution time : 336.150 s
Press any key to continue.

```

5. Write C program to search element in an array.

Input:

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

```
int main(){
```

```
    int i,n,value,pos=-1;
```

```

printf("Enter size of the array: ");
scanf("%d",&n);
int arr[n];
for(i=0;i<n;i++){
    scanf("%d",&arr[i]);
}
printf("Enter number you want to searce:\n ");
scanf("%d",&value);
for(i=0;i<n;i++){
    if(value==arr[i]) {
        pos=i+1;
        break;
    }
}
if(pos==-1){
    printf("item is not found") }
else{
    printf("The value is found at %d",pos); }
return 0;
}

```

Output:

```

Enter size of the array: 10
1 2 3 4 5 6 7 8 9 10
Enter number you want to searce:
3
The value is found at 3
Process returned 0 (0x0)   execution time : 15.326 s
Press any key to continue.

```

