DAA – Practical 2

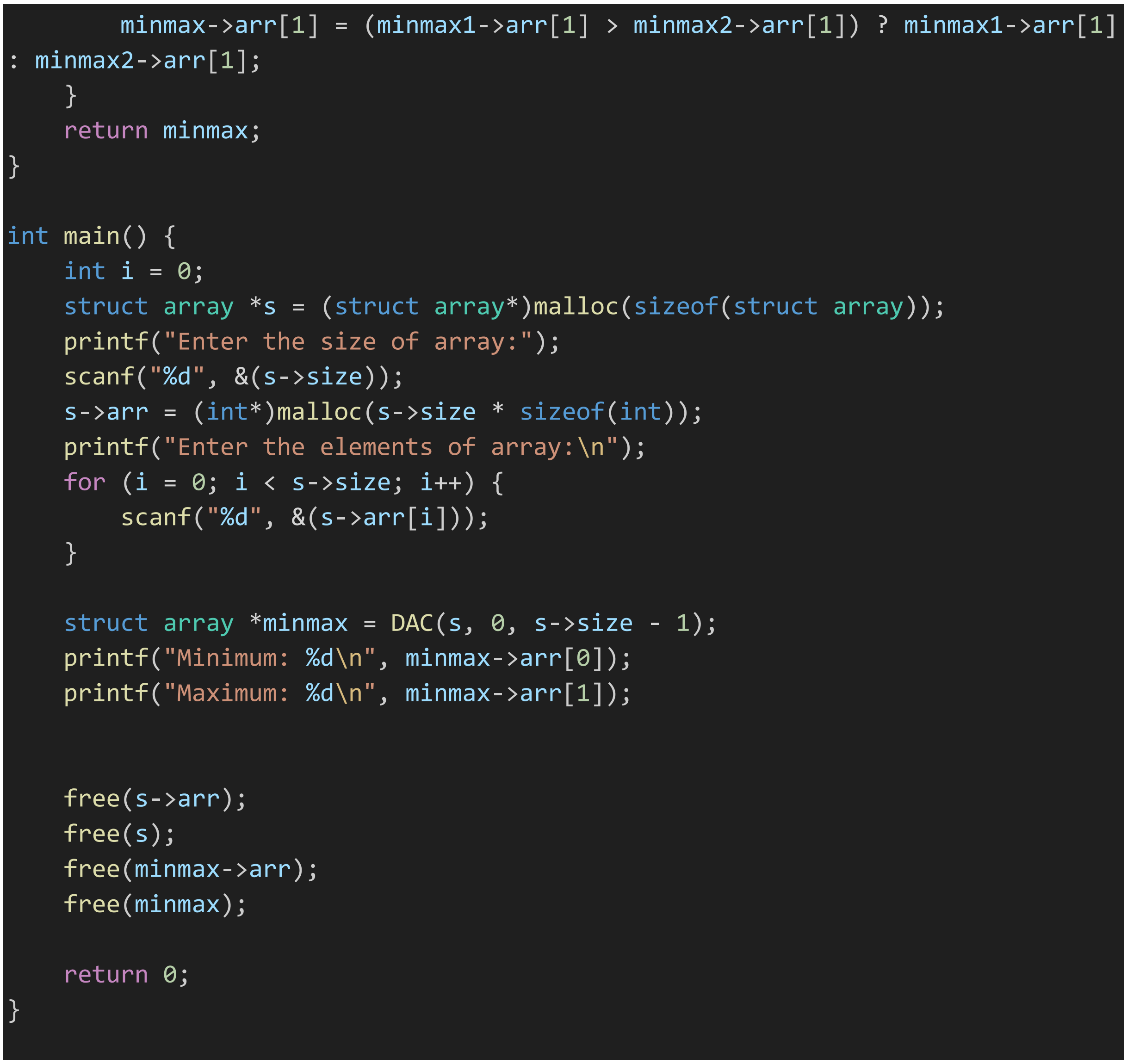
Name: Aditya.Rajesh.Wanwade

Roll no :27 && Batch:C2

2.a) Min Max using divide and conquer approach

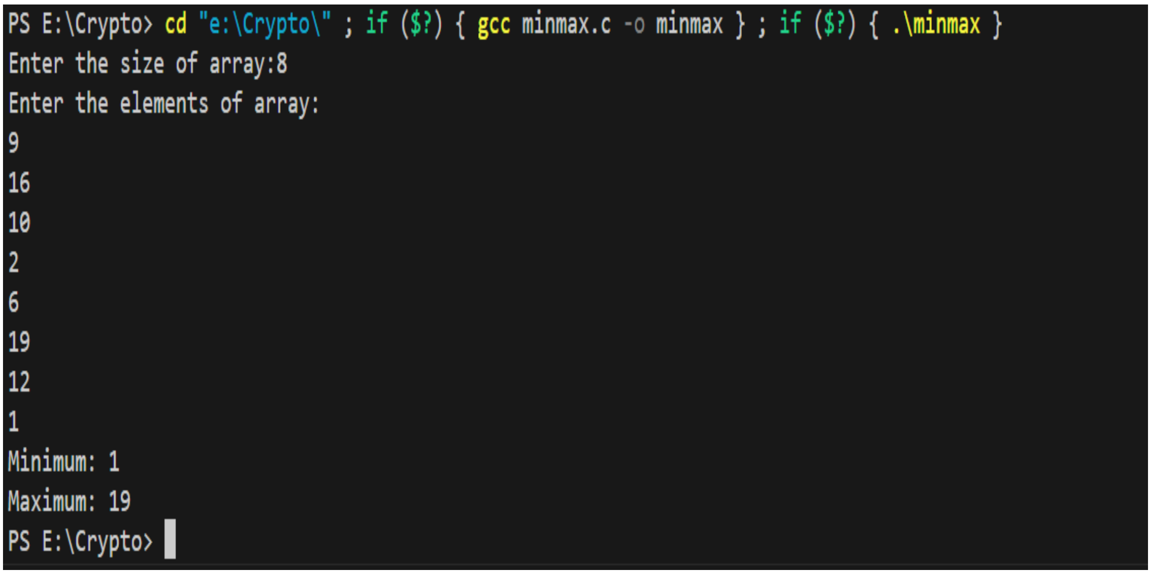
Code of Minmax:

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

struct array {
    int size;
    int *arr;
};

struct array* DAC(struct array *s, int l, int r) {
    struct array *minmax = (struct array*)malloc(sizeof(struct array));
    minmax->size = 2;
    minmax->arr = (int*)malloc(minmax->size * sizeof(int));

    int min, max;
    if (l == r) {
        minmax->arr[0] = s->arr[l];
        minmax->arr[1] = s->arr[r];
    } else if (l == r - 1) {
        if (s->arr[l] > s->arr[r]) {
            minmax->arr[0] = s->arr[r];
            minmax->arr[1] = s->arr[l];
        } else {
            minmax->arr[0] = s->arr[l];
            minmax->arr[1] = s->arr[r];
        }
    } else {
        int m = l + (r - l) / 2;

        struct array *minmax1 = DAC(s, l, m);
        struct array *minmax2 = DAC(s, m + 1, r);

        minmax->arr[0] = (minmax1->arr[0] < minmax2->arr[0]) ? minmax1->arr[0] : minmax2->arr[0];
```



```
        minmax->arr[1] = (minmax1->arr[1] > minmax2->arr[1]) ? minmax1->arr[1]
: minmax2->arr[1];
    }
    return minmax;
}

int main() {
    int i = 0;
    struct array *s = (struct array*)malloc(sizeof(struct array));
    printf("Enter the size of array:");
    scanf("%d", &(s->size));
    s->arr = (int*)malloc(s->size * sizeof(int));
    printf("Enter the elements of array:\n");
    for (i = 0; i < s->size; i++) {
        scanf("%d", &(s->arr[i]));
    }

    struct array *minmax = DAC(s, 0, s->size - 1);
    printf("Minimum: %d\n", minmax->arr[0]);
    printf("Maximum: %d\n", minmax->arr[1]);


    free(s->arr);
    free(s);
    free(minmax->arr);
    free(minmax);

    return 0;
}
```

Output:

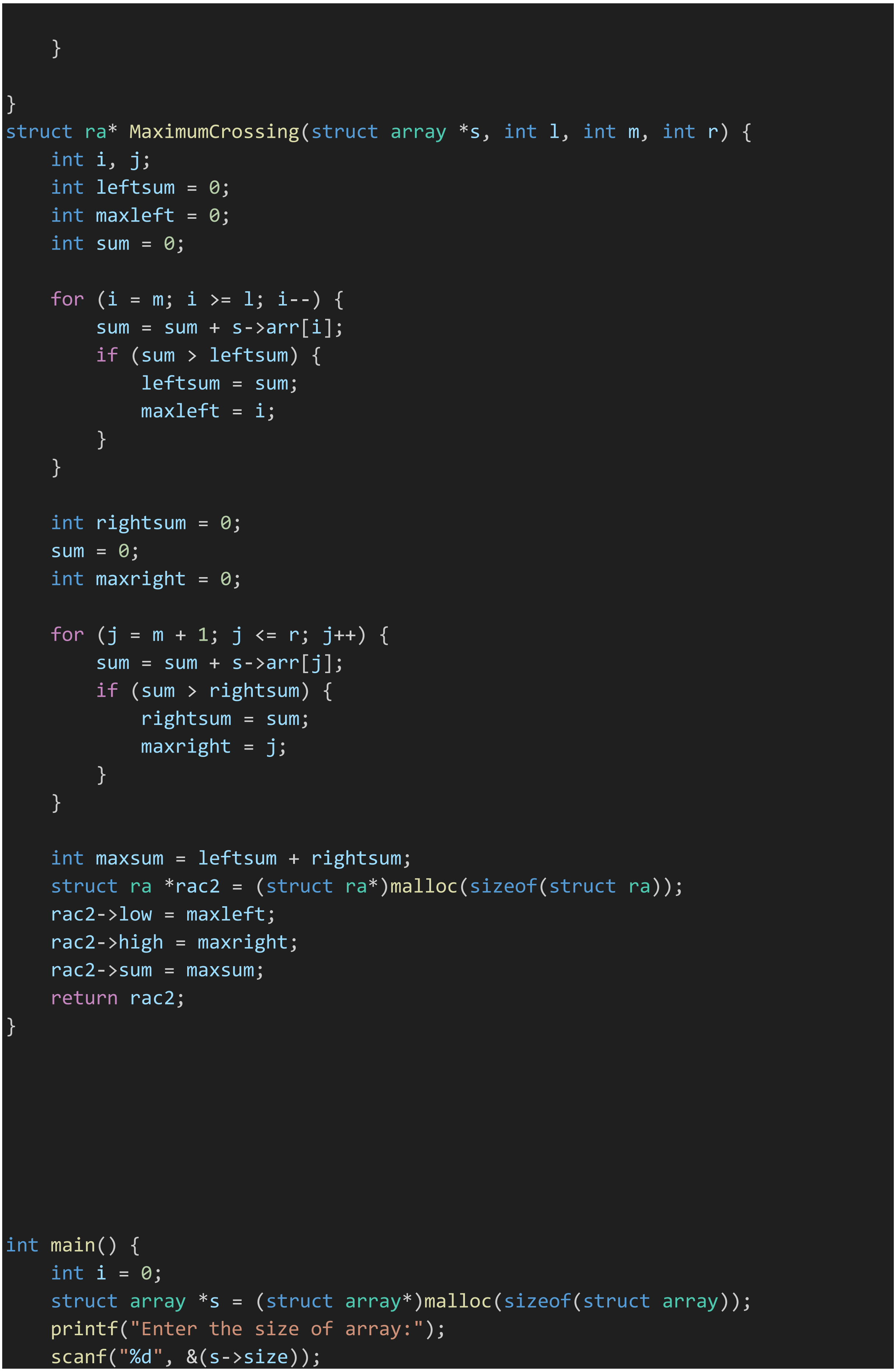


```
PS E:\Crypto> cd "e:\Crypto\" ; if ($?) { gcc minmax.c -o minmax } ; if ($?) { .\minmax }
Enter the size of array:8
Enter the elements of array:
9
16
10
2
6
19
12
1
Minimum: 1
Maximum: 19
PS E:\Crypto>
```

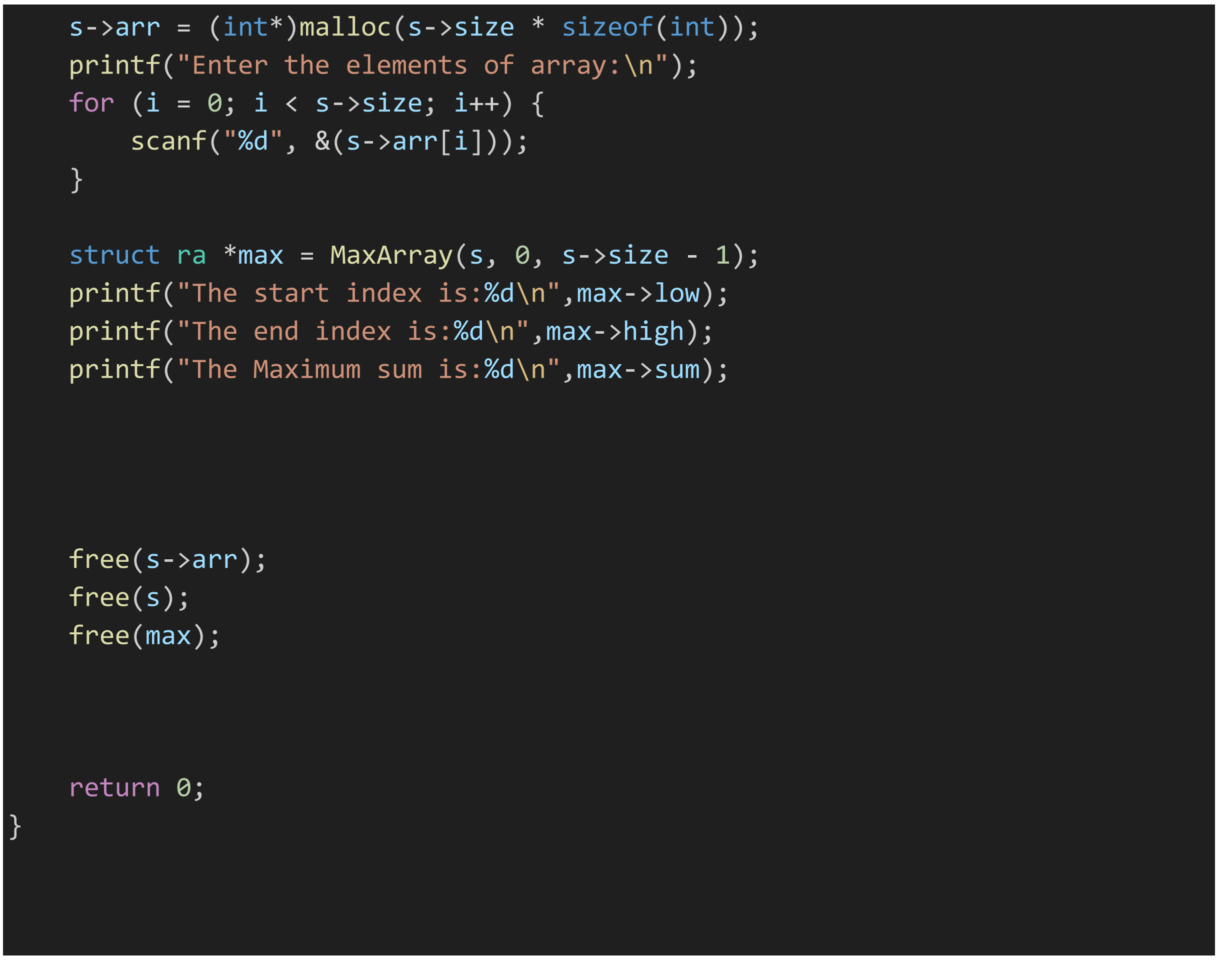
2.b) Maximum subarray problem code for given array elements:

Code of Maximum Subarray:

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
struct array {
    int size;
    int *arr;
};
struct ra{
int low;
int high;
int sum;
};
struct ra* MaximumCrossing(struct array *s, int l, int m, int r);

struct ra* MaxArray(struct array *s, int l, int r) {

struct ra *rac1=(struct ra*)malloc(sizeof(struct ra));
    int min, max;
    if (l == r) {
        rac1->low=l;
        rac1->high=r;
        rac1->sum=s->arr[l];
        return rac1;
    }
    else{


        int m = l + (r - l) / 2;

        struct ra *ra1 = MaxArray(s, l, m);
        struct ra *ra2 = MaxArray(s, m + 1, r);
        struct ra *rac=MaximumCrossing(s,l,m,r);
        if(ra1->sum>=ra2->sum && ra1->sum>=rac->sum){
            return ra1;
        }
        else if(ra2->sum>=ra1->sum && ra2->sum>=rac->sum){
            return ra2;
        }
        else{
            return rac;
        }
```

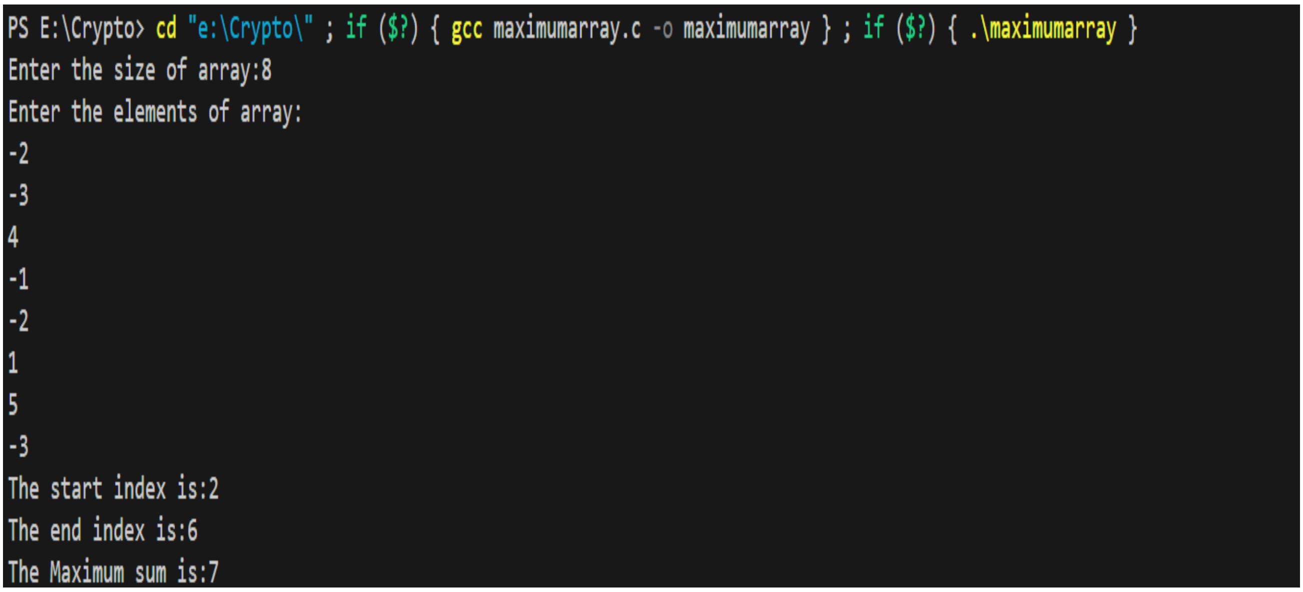


```
    }

}
struct ra* MaximumCrossing(struct array *s, int l, int m, int r) {
    int i, j;
    int leftsum = 0;
    int maxleft = 0;
    int sum = 0;

    for (i = m; i >= l; i--) {
        sum = sum + s->arr[i];
        if (sum > leftsum) {
            leftsum = sum;
            maxleft = i;
        }
    }

    int rightsum = 0;
    sum = 0;
    int maxright = 0;

    for (j = m + 1; j <= r; j++) {
        sum = sum + s->arr[j];
        if (sum > rightsum) {
            rightsum = sum;
            maxright = j;
        }
    }

    int maxsum = leftsum + rightsum;
    struct ra *rac2 = (struct ra*)malloc(sizeof(struct ra));
    rac2->low = maxleft;
    rac2->high = maxright;
    rac2->sum = maxsum;
    return rac2;
}






int main() {
    int i = 0;
    struct array *s = (struct array*)malloc(sizeof(struct array));
    printf("Enter the size of array:");
    scanf("%d", &(s->size));
```



```c
    s->arr = (int*)malloc(s->size * sizeof(int));
    printf("Enter the elements of array:\n");
    for (i = 0; i < s->size; i++) {
        scanf("%d", &(s->arr[i]));
    }

    struct ra *max = MaxArray(s, 0, s->size - 1);
    printf("The start index is:%d\n",max->low);
    printf("The end index is:%d\n",max->high);
    printf("The Maximum sum is:%d\n",max->sum);




    free(s->arr);
    free(s);
    free(max);



    return 0;
}
```

Output:



```
PS E:\Crypto> cd "e:\Crypto\" ; if ($?) { gcc maximumarray.c -o maximumarray } ; if ($?) { .\maximumarray }
Enter the size of array:8
Enter the elements of array:
-2
-3
4
-1
-2
1
5
-3
The start index is:2
The end index is:6
The Maximum sum is:7
```