```c
/*### function to solve the maximum subarray problem #####*/
#include <stdio.h>

typedef struct
{
    int start;
    int end;
    int sum;
} Subarray;

Subarray findMaxSubarray(int arr[], int size)
{
    Subarray current ={0};
    Subarray max ={0};
    current.sum = arr[0];
    max.sum = arr[0];

    for (int i = 1; i < size; i++)
    {
        if (arr[i] > current.sum + arr[i])
        {
            current.start = current.end = i;
            current.sum = arr[i];
        }
        else
        {
            current.end = i;
            current.sum += arr[i];
        }

        if (current.sum > max.sum)
        {
            max.start = current.start;
            max.end = current.end;
            max.sum = current.sum;
        }
    }

    return max;
}

int main()
{
    int arr[] = {-2, 1, -3, 4, -1, 2, 3, -5, 4};
    int size = sizeof(arr) / sizeof(arr[0]);

    Subarray result = findMaxSubarray(arr, size);

    printf("Maximum subarray sum: %d\n", result.sum);
    printf("Start index: %d, End index: %d\n", result.start, result.end);

    return 0;
}

/*
This implementation defines a `Subarray` structure to store the start and end
indices of the maximum subarray found so far, along with its sum. The
`findMaxSubarray` function iterates through the array while keeping track of the
current subarray and the maximum subarray. The `main` function demonstrates the
```

```
usage of the `findMaxSubarray` function with the given example array.*/
```