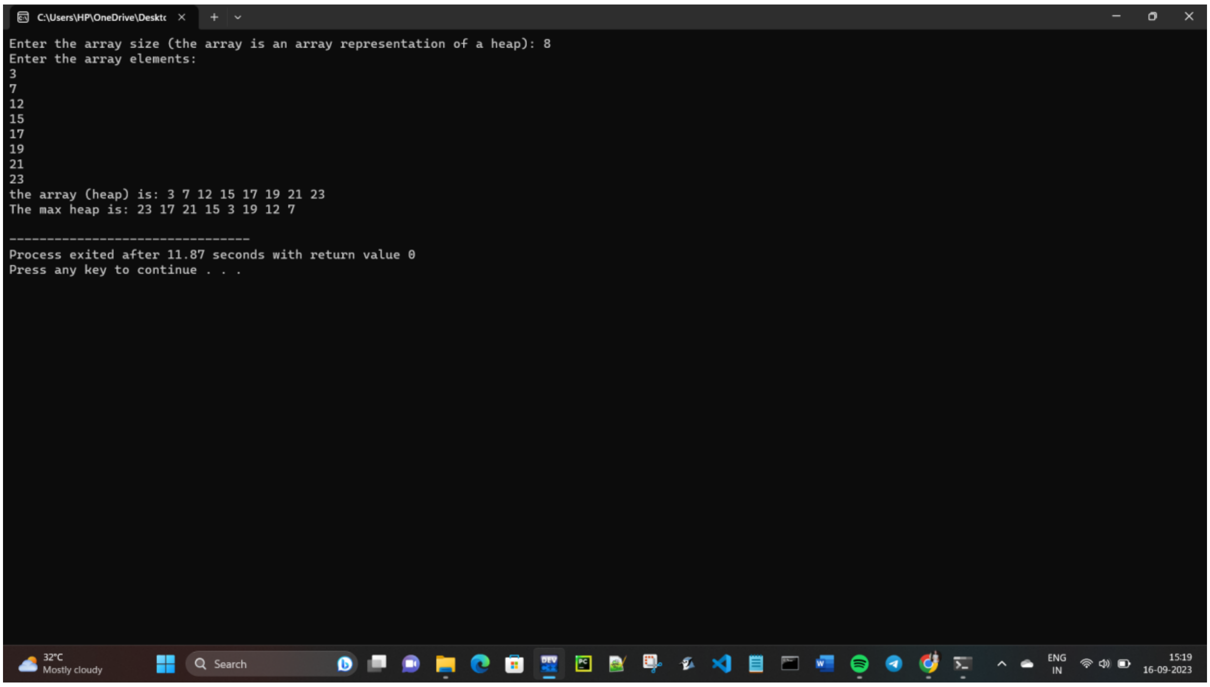
```
/*max heapify*/
#include <stdio.h>

void create_max_heap(int arr[], int n);
void max_heapify(int arr[], int n, int i);

int main() {
    int n, i;
    printf("Enter the array size (the array is an array representation of a heap): ");  //so,heap size=arr size
    scanf("%d", &n);
    int arr[n];
    printf("Enter the array elements:\n");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("the array (heap) is: ");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    create_max_heap(arr, n);
    printf("The max heap is: ");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
    return 0;
}

void create_max_heap(int arr[], int n) {
        //here heap size=arr size to show that we are doing max heapify in entire max heap
    int last_non_leaf = (n - 1) / 2;   //last_non_leaf==largest non leaf index,id starting from 0
    for (int i = last_non_leaf; i >= 0; i--) {
        max_heapify(arr, n, i);  ////i=last_non_leaf
    }
}

void max_heapify(int arr[], int n, int i) {
    int lc, rc, largest;
    lc = 2 * i + 1;
    rc = 2 * i + 2;
    if (lc < n && arr[lc] > arr[i]) {  //here arr.heapsize=n=arr.length as heap size=arr size
        largest = lc;
    } else {
        largest = i;
    }
    if (rc < n && arr[rc] > arr[largest]) {  //arr[rc] is larger then the largest element determined in just
the prv if loop,ie arr[rc] is largest among 3 nodes(if have 3 nodes)
        largest = rc;
    }
    if (largest != i) {  //swapp arr[i] with arr[largest]
        int temp = arr[i];
        arr[i] = arr[largest];
        arr[largest] = temp;
        max_heapify(arr, n, largest);  //Recursively heapify the affected sub-tree,subtree with root as
largest
        //we are sending the lagest id as it may happen that after doing max heapify any child of the
largest is larger then it so we need to apply max heapify there so that the max heap will follow the
max heap property again
}
}
```



```
Enter the array size (the array is an array representation of a heap): 8
Enter the array elements:
3
7
12
15
17
19
21
23
the array (heap) is: 3 7 12 15 17 19 21 23
The max heap is: 23 17 21 15 3 19 12 7

--------------------------------
Process exited after 11.87 seconds with return value 0
Press any key to continue . . .
```