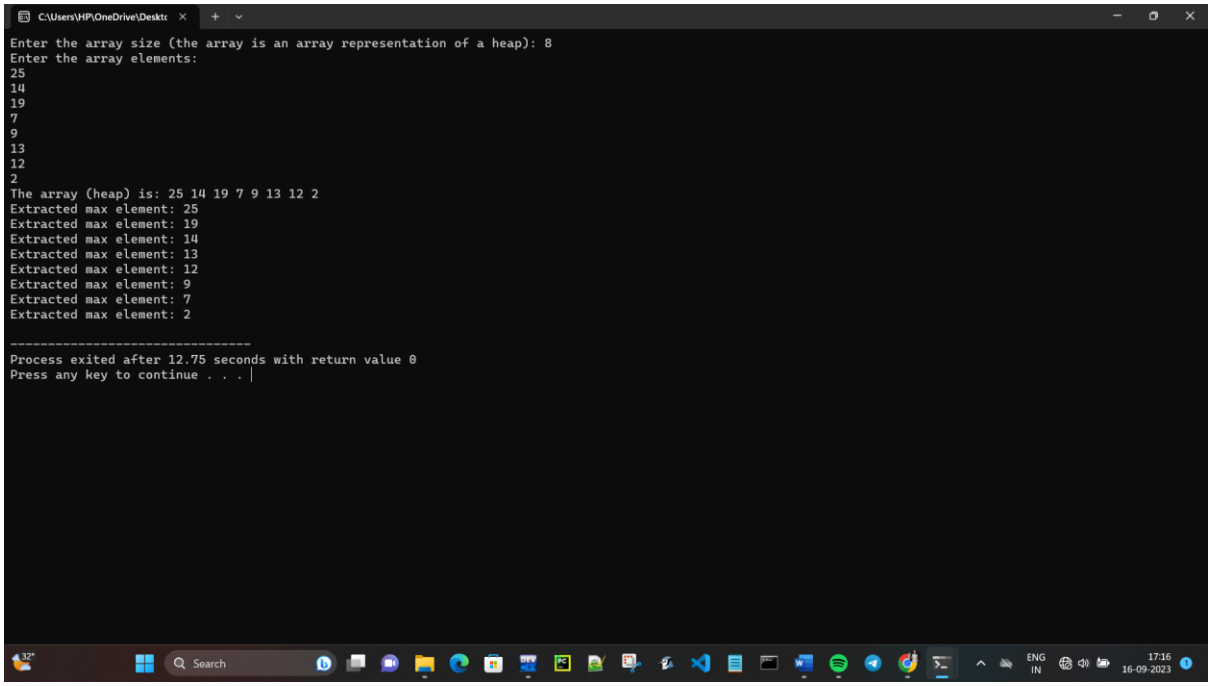
/*extract max element from max heap*/

```
#include <stdio.h>
int extract_max_ele(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");
   while (n > 0) {    //as,if n=0 then no element in heap
      int max = extract_max_ele(arr, n);
      printf("Extracted max element: %d\n", max);
   }
   return 0;
}
int extract_max_ele(int arr[],int n){
            if(n<0){
                        printf("ERROR:heap underflow,there is no element in that heap\n");
            }
            int max=arr[0];
            arr[0]=arr[n-1];   //as index is starting from 0
```

```
            n--;
            max_heapify(arr,n,0);  //heapifying from the 1st index
            return max;
}
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
}
}
```


C:\Users\HP\OneDrive\Deskto
Enter the array size (the array is an array representation of a heap): 8
Enter the array elements:
25
14
19
7
9
13
12
2
The array (heap) is: 25 14 19 7 9 13 12 2
Extracted max element: 25
Extracted max element: 19
Extracted max element: 14
Extracted max element: 13
Extracted max element: 12
Extracted max element: 9
Extracted max element: 7
Extracted max element: 2

Process exited after 12.75 seconds with return value 0
Press any key to continue . . .
32°
Search
ENG
IN
17:16
16-09-2023