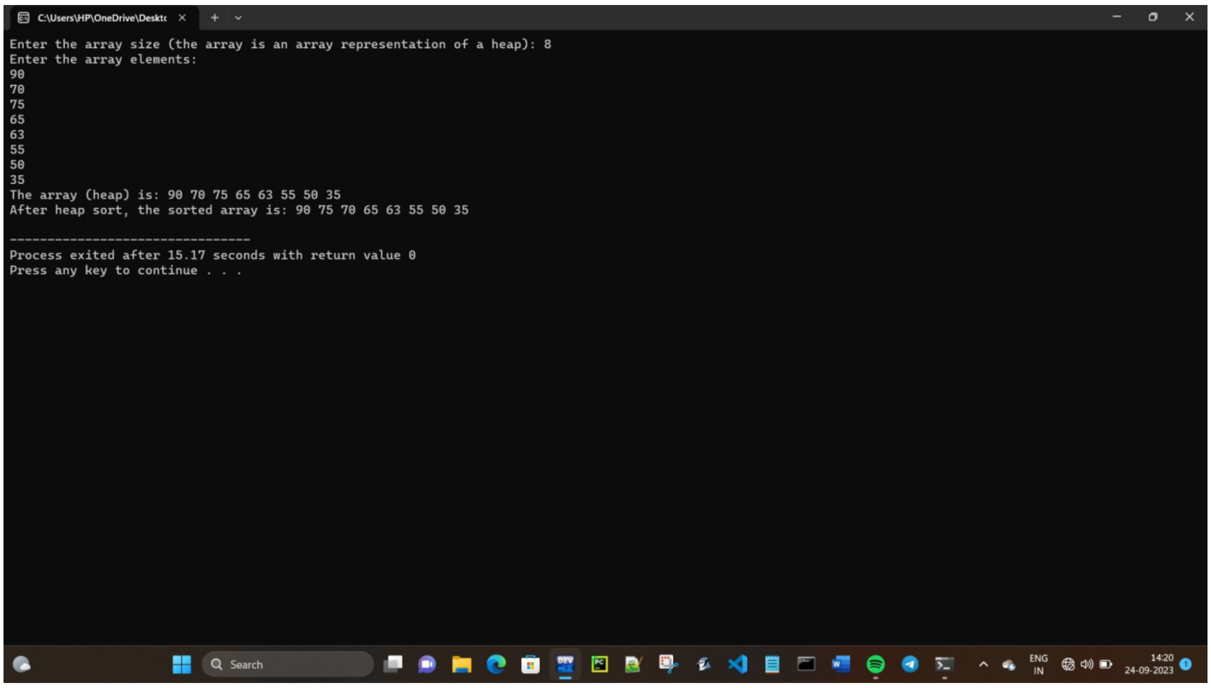
/*heap sort(descending)sir*/

```
#include <stdio.h>
int n = 0;
void heap_sort(int arr[]);
void create_max_heap(int arr[]);
void max_heapify(int arr[], int n, int i);
int main() {
int i, arr[20];
printf("Enter the array size (the array is an array representation of a heap): ");
scanf("%d", &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");
heap_sort(arr);
printf("After heap sort, the sorted array is: ");
for (i = 0; i < n; i++) {
printf("%d ", arr[i]);
}
printf("\n");
return 0;
}
void heap_sort(int arr[]) {
int copy = n;  //n=size
create_max_heap(arr);
for (int i = n - 1; i >= 1; i--) {
```

```
int max = arr[0];

arr[0] = arr[i];

arr[i] = max;

n--;

max_heapify(arr, n, 0);

}

n=copy;

}

void create_max_heap(int arr[]) {

int last_non_leaf = (n - 1) / 2;

for (int i = last_non_leaf; i >= 0; i--) {

max_heapify(arr, n, i);

}

}

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]) {

largest = lc;

} else {

largest = i;

}

if (rc < n && arr[rc] > arr[largest]) {

largest = rc;

}

if (largest != i) {

int temp = arr[i];arr[i] = arr[largest];

arr[largest] = temp;

max_heapify(arr, n, largest);

}
```

}



```
Enter the array size (the array is an array representation of a heap): 8
Enter the array elements:
90
70
75
65
63
55
50
35
The array (heap) is: 90 70 75 65 63 55 50 35
After heap sort, the sorted array is: 90 75 70 65 63 55 50 35

--------------------------------
Process exited after 15.17 seconds with return value 0
Press any key to continue . . .
```