/*heap sort*/

```
#include <stdio.h>

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

int main() {
   int i, n;
   printf("Enter the array size (the array is an array representation of a heap): ");
   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");
   heap_sort(arr, n);
   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 n) {
```

```
    create_max_heap(arr, n);
    for (int i = n - 1; i >= 1; i--) {
        int max = arr[0];
        arr[0] = arr[i];
        arr[i] = max;
        max_heapify(arr, i, 0);
    }
}

void create_max_heap(int arr[], int n) {
    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);
    }
}
```