//heap sort(replacement)

```
#include<stdio.h>
int n=0;

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

int main(){
	int arr[20],i;
	printf("enter the size of array(heap): ");
	scanf("%d",&n);
	printf("\nenter the array eles\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]);
	}
	heap_sort(arr);
	printf("\nafter heap sort the heap(max heap) is: ");
	for(i=n-1;i>=0;i--){
		printf("%d ",arr[i]);
	}
	printf("\n");
}

void heap_sort(int arr[]){
	int copy=n;
	create_max_heap(arr);
```

```
	for(int i=n-1;i>=1;i--){  //as when n=2 then arr[0] will be swapped with arr[1](as it is a mh so arr[0]>arr[1],arr[1]=2nd last smallest,arr[0]=smallest after heap sort),when n=1 then there are no eles to compare with arr[root] so the loop will run till 1
		int max=arr[0];
		arr[0]=arr[i];
		arr[i]=max;
		n--;   //now the loop will not count the max and heapify the full heap from root till n-1(n-1 doesnt includes the max)
		max_heapify(arr,0);
	}
	n=copy;     //as n-- was globally changed
}

void create_max_heap(int arr[]){
	int largest_non_leaf=(n-1)/2;
	for(int i=largest_non_leaf;i>=0;i--){
		max_heapify(arr,i);
	}
}

void max_heapify(int arr[],int i){
	int rc,lc,largest;
	lc=2*i+1;
	rc=2*i+2;
	if(lc<n && arr[i]<arr[lc]){
		largest=lc;
	}
	else{
		largest=i;
	}
	if(rc<n && arr[largest]<arr[rc]){
		largest=rc;
```

```
        }
        if(largest!=i){
                int temp=arr[i];
                arr[i]=arr[largest];
                arr[largest]=temp;
                max_heapify(arr,largest);
        }
}
```