#include<stdio.h>

#include<stdlib.h>

#include<stdbool.h>

void swap(int \*a,int \*b)

{

int temp=\*a;

\*a=\*b;

\*b=temp;

}

void inc\_key(int heap[],int ind)

{

int p=(ind-1)/2;

while(ind && heap[p]<heap[ind])

{

swap(&heap[p],&heap[ind]);

ind=p;

p=(ind-1)/2;

}

}

void maxheapify(int heap[],int i,int size)

{

int lc=(2\*i)+1,rc=(2\*i)+2,largest=i,temp;

if(lc<size && heap[lc]>heap[largest])

{

largest=lc;

}

if(rc<size && heap[rc]>heap[largest])

{

largest=rc;

}

if(largest!=i)

{

swap(&heap[largest],&heap[i]);

maxheapify(heap,largest,size);

}

}

void sort(int heap[],int size)

{

while(size)

{

maxheapify(heap,0,size--);

swap(&heap[0],&heap[size]);

}

}

void create\_heap(int heap[],int size)

{

int i;

for(i=(size-2)/2;i>=0;i--)

{

maxheapify(heap,i,size);

}

}

bool check(int heap[],int size,bool \*isheap)

{

int ch;

printf("1.Create Heap and Any other key to Abort Operation\n");

scanf("%d",&ch);

if(ch==1)

{

create\_heap(heap,size);

\*isheap=true;

return true;

}

return false;

}

int main()

{

int ch=1,size=0,i,j,\*heap ;

bool isheap=true;

while(ch)

{

printf("1.Enqueue 2.Dequeue 3.Change Key 4.Front Element 5.Sort 6.Display 0.Exit\n");

scanf("%d",&ch);

if(ch==1)

{

if(!isheap && !check(heap,size,&isheap))

{

continue;

}

else if(size==0)

{

size++;

heap=(int\*)malloc(sizeof(int));

printf("Enter Data:");

scanf("%d",&heap[size-1]);

}

else

{

size++;

heap=(int\*)realloc(heap,sizeof(int)\*size);

printf("Enter Data:");

scanf("%d",&heap[size-1]);

inc\_key(heap,size-1);

}

}

else if(ch==2)

{

if(!size || heap==NULL)

{

printf("Empty Queue\n");

}

else if(!isheap && !check(heap,size,&isheap))

{

continue;

}

else

{

printf("Removing %d\n",heap[0]);

heap[0]=heap[--size];

heap=(int\*)realloc(heap,sizeof(int)\*size);

maxheapify(heap,0,size);

}

}

else if(ch==3)

{

if(!isheap && !check(heap,size,&isheap))

{

continue;

}

else if(size==0 || heap==NULL)

{

printf("Empty Queue\n");

}

else

{

int ok,nk;

printf("Enter Old Key & New Key\n");

scanf("%d%d",&ok,&nk);

for(i=0;i<size;i++)

{

if(heap[i]==ok)

{

heap[i]=nk;

break;

}

}

if(i==size)

{

printf("Old Key Not Found\n");

}

else

{

if(nk>ok)

{

inc\_key(heap,size-1);

}

else if(nk<ok)

{

maxheapify(heap,i,size);

}

}

}

}

else if(ch==4)

{

if(!isheap && !check(heap,size,&isheap))

{

continue;

}

if(!size || heap==NULL)

{

printf("Empty Queue\n");

}

else

{

printf("Front Element is %d\n",heap[0]);

}

}

else if(ch==5)

{

if(!size || heap==NULL)

{

printf("Empty Queue\n");

}

else

{

isheap=false;

sort(heap,size);

}

}

else if(ch==6)

{

if(size==0 || heap==NULL)

{

printf("Empty Queue\n");

}

else

{

for(i=0;i<size;i++)

{

printf("%d ",heap[i]);

}

printf("\n");

}

}

else if(ch)

{

printf("Wrong Input\n");

}

}

}