```
//Name: Mehmet Fatih Çelik
//ID: 2385268

//I used the algorithm that Meryem hoca gave us.

#include <stdio.h>
#include <stdlib.h>

struct HeapStruct{
	int capacity , size;
	char elements[50];
};

void Insert(int [], struct HeapStruct *);
int DeleteMin(struct HeapStruct *);

int main(){
	struct HeapStruct *heap;
	heap = (struct HeapStruct*)malloc(sizeof(struct HeapStruct));
	if (heap == NULL){
		printf("Error occured while allocating the memory!");
		exit(-1);
	}
	heap->size = 0;
	heap->capacity = 20;

	FILE *inFile;
	inFile = fopen("file.txt","r");
	if (inFile == NULL){
		printf("Error occured while reading the file!");
		exit(-1);
```

```
	}


	int nums[15];
	while(fscanf(inFile,"%d,%d,%d,%d,%d,%d,%d",&nums[0],&nums[1],&nums[2],&nums[3],&nums[4],&nums[5],&nums[6]) != EOF)
		printf("Nums: %d %d %d %d %d %d %d",nums[0],nums[1],nums[2],nums[3],nums[4],nums[5],nums[6]);


	fclose(inFile);


	int minElement;
	Insert(nums, heap);
	minElement = DeleteMin(heap);



	return 0;
}


void Insert(int nums[], struct HeapStruct *heap){
	int i, j = 0;
	if(heap->size == heap->capacity){
		printf("The heap is full!");
		exit(-1);
	}


	for(i= ++heap->size; heap->elements[i/2] > nums[j]; i/=2){
		heap->elements[i] = heap->elements[i/2];
		j++;
		heap->size++;
	}


	heap->size -= 4;
```

```
}

int DeleteMin(struct HeapStruct *heap){
	int i, child;
	int minElement, lastElement;
	if(heap->size == 0){
		printf("Heap is empty!");
		exit(-1);
	}

	minElement = heap->elements[1];
	lastElement = heap->elements[heap->size--];

	for(i=1; i*2 <= heap->size; i=child){
		//smaller child finding
		child *= 2;

		if((child != heap->size) && (heap->elements[child+1] < heap->elements[child]))
			child++;

		//percolating
		if(lastElement > heap->elements[child])
			heap->elements[i] = heap->elements[child];
		else
			break;
	}

	heap->elements[i] = lastElement;
	return minElement;
}
```