//GAYATHRI.M-185001050

// HEAP-PRIORITY QUEUE

Program:
Contents of functions.h file

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

typedef struct employee{
	char name[30];
	int id;
	float salary;
}emp;

emp getemp(){
	emp e;

	printf("\nEnter the name   : ");
	scanf("%[^\n]",e.name);
	printf("Enter the id     : ");
	scanf("%d",&e.id);
	printf("Enter the salary : ");
	scanf("%f",&e.salary);
	getchar();
	printf("\n");
	return e;
}

void putemp(const emp e){
	printf("Name     : %s\n",e.name);
	printf("ID       : %d\n",e.id);
	printf("Salary   : %.2f\n",e.salary);
}

typedef struct PriorityQueue{
	int capacity;
	int size;
	emp* arr;
}*PQueue;

int isFull(PQueue Q){
	return Q -> size == Q -> capacity;
}

int isEmpty(PQueue Q){
	return Q -> size == 0;
}


PQueue createPQueue(const int maxsize){
	PQueue tmp = (PQueue)malloc(sizeof(PQueue));

	tmp -> capacity = maxsize;
	tmp -> size = 0;
	tmp -> arr = (emp*)malloc(sizeof(emp) * maxsize);

	tmp -> arr[0].salary = 999999.9;
	return tmp;
}

void enqueue(PQueue q,const emp d){
	if(isFull(q)){
		printf("Queue Full!\n");
		return;
	}
	int i = ++q -> size;
	for(; q -> arr[i/2].salary < d.salary ; i /= 2)
		q -> arr[i] = q -> arr[i/2];

	q -> arr[i] = d;

}
```

```
emp dequeue(PQueue q){
        if(isEmpty(q)){
                printf("Queue Empty!\n");
                return q -> arr[0];
        }
        int i,child;
        emp min,last;

        min = q -> arr[1];
        last = q -> arr[q -> size--];

        for(i = 1; i * 2 <= q -> size ; i = child){
                child = i * 2;

                if(child != q -> size && q -> arr[child + 1].salary > q ->
arr[child].salary)
                        child ++;
                if(last.salary < q -> arr[child].salary)
                        q -> arr[i] = q -> arr[child];
                else
                        break;
        }

        q -> arr[i] = last;
        return min;
}

void display(PQueue Q){
        for(int i = 1 ; i <= Q -> size ; i++)
                putemp(Q -> arr[i]);
}
```

Contents of main.c file

```
#include "functions.h"

int main(void){
        PQueue q = createPQueue(10);

        for(int i = 0 ; i < 5 ; i++){
                enqueue(q,getemp());
                printf("Queue after adding: \n");
                display(q);
        }
        printf("De-Queued Element\n");
        putemp(dequeue(q));
}
```

Output:

```
Enter the name   : Sandra
Enter the id     : 123
Enter the salary : 40000

Queue after adding:
Name     : Sandra
ID       : 123
Salary   : 40000.00

Enter the name   : Aira
Enter the id     : 156
Enter the salary : 65000

Queue after adding:
Name     : Aira
ID       : 156
Salary   : 65000.00
Name     : Sandra
ID       : 123
Salary   : 40000.00

Enter the name   : Banu
Enter the id     : 124
Enter the salary : 59000
```

```
Queue after adding:
Name     : Aira
ID       : 156
Salary   : 65000.00
Name     : Sandra
ID       : 123
Salary   : 40000.00
Name     : Banu
ID       : 124
Salary   : 59000.00

Enter the name   : Tina
Enter the id     : 192
Enter the salary : 70000

Queue after adding:
Name     : Tina
ID       : 192
Salary   : 70000.00
Name     : Aira
ID       : 156
Salary   : 65000.00
Name     : Banu
ID       : 124
Salary   : 59000.00
Name     : Sandra
ID       : 123
Salary   : 40000.00

Enter the name   : Sam
Enter the id     : 164
Enter the salary : 80000

Queue after adding:
Name     : Sam
ID       : 164
```

```
Salary   : 80000.00
Name     : Tina
ID       : 192
Salary   : 70000.00
Name     : Banu
ID       : 124
Salary   : 59000.00
Name     : Sandra
ID       : 123
Salary   : 40000.00
Name     : Aira
ID       : 156
Salary   : 65000.00

De-Queued Element

Name     : Sam
ID       : 164
Salary   : 80000.00
```