```
//S Harshini-185001058

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

typedef struct details{
        char name[50];
        int id;
        float salary;
}details;

typedef struct heap{
   int capacity;
   int size;
   details *d;
}priorityQueue;

priorityQueue* Create(int max){
   priorityQueue *pq;
   pq=(priorityQueue*)malloc(sizeof(priorityQueue));
   if(pq==NULL){
      printf("Out of space");
      return NULL;
   }
   pq->d=(details*)malloc((max+1)*sizeof(details));
   pq->capacity=max;
   pq->size=0;
   pq->d[0].salary=50000.0;
   return pq;
}

void Insert(details data,priorityQueue *pq){
   int i;
   if(pq->size==pq->capacity){
      printf("PQueue is Full");
      return;
   }
   for(i=++pq->size; pq->d[i/2].salary < data.salary;i/=2){
      pq->d[i]=pq->d[i/2];
   }
   pq->d[i] =data;
}
```

```
void DeleteMin(priorityQueue *pq){
    int i,child;
    details min,last;
    if(pq->size==0){
        printf("PQueue is Empty\n");
    }
    min=pq->d[0];
    last=pq->d[pq->size--];
    for(i=1; (i*2)<pq->size;i=child){
        child=i*2;
        if(child!=pq->size && pq->d[child+1].salary > pq->d[child].salary)
            child++;
        if(last.salary < pq->d[child].salary)
            pq->d[i]=pq->d[child];
        else
            break;
    }
    pq->d[i]=last;
}

void Display(priorityQueue *pq){
    printf("\nTree:\n");
    for(int i=1;i<=pq->size;i++)
        printf("Name:%s Id:%d Salary:%f\n",pq->d[i].name,pq->d[i].id,pq->d[i].salary);
    printf("\n");
}

int main(){
    priorityQueue *pq=Create(100);
    int choice,loop=1;
    char name[50];
    int id;
    float sal;
    details d;
    do{
        printf("\nenter choice \n1)Insert 2)relieve 3)Display 4)quit\nEnter Choice:");
        scanf("%d",&choice);
        switch(choice){
            case 1:
                printf("Enter name,id,sal:\n");
                scanf("%s",name);
                scanf("%d",&id);
```

```
            scanf("%f",&sal);
                strcpy(d.name,name);
                d.id=id;
                d.salary=sal;
            Insert(d,pq);
                Display(pq);
            break;
        case 2:
                DeleteMin(pq);
                Display(pq);
            break;
        case 3:
            Display(pq);
            break;
        case 4:
            loop=0;
      }
   }while(loop);
}


/*SAMPLE INPUT/OUTPUT

enter choice
1)Insert 2)relieve 3)Display 4)quit
Enter Choice:1
Enter name,id,sal:
harshu
1
50000

Tree:
Name:harshu Id:1 Salary:50000.000000


enter choice
1)Insert 2)relieve 3)Display 4)quit
Enter Choice:2

Tree:


enter choice
```

1)Insert 2)relieve 3)Display 4)quit
Enter Choice:1
Enter name,id,sal:
archu
2
6000

Tree:
Name:archu Id:2 Salary:6000.000000

enter choice
1)Insert 2)relieve 3)Display 4)quit
Enter Choice:1
Enter name,id,sal:
jack
3
7000

Tree:
Name:jack Id:3 Salary:7000.000000
Name:archu Id:2 Salary:6000.000000

enter choice
1)Insert 2)relieve 3)Display 4)quit
Enter Choice:3

Tree:
Name:jack Id:3 Salary:7000.000000
Name:archu Id:2 Salary:6000.000000

enter choice
1)Insert 2)relieve 3)Display 4)quit
Enter Choice:2

Tree:
Name:archu Id:2 Salary:6000.000000

enter choice
1)Insert 2)relieve 3)Display 4)quit

Enter Choice:4
*/