**Ex-10 Priority queue using binary heap**

-------------------------------------------------------------------------------------------------------------------------------

**Main**

#include <stdio.h>

#include <stdlib.h>

#define MAXDATA 99999999

#include "priorityheap.h"

int main(){

int maxele;

printf("Enter the max number of element in the heap: ");

scanf("%d", &maxele);

heap \*pqueue;

pqueue = init(maxele);

heapdatatype data;

int choice;

printf("PRIORITY QUEUE OPERATIONS\n1: Insert\n2: Delete\n3: Display\n");

printf("\nEnter your choice: ");

scanf("%d",&choice);

do{

switch(choice){

case 1: printf("\nEnter the employee's name: ");

scanf("%s", data.name);

printf("Enter the employee's id: ");

scanf("%d", &data.id);

printf("Enter the employee's salary: ");

scanf("%d", &data.salary);

insert(data,pqueue);

break;

case 2: deletemin(pqueue);

break;

case 3: display(pqueue->size,pqueue);

break;

default: printf("\nEnter a valid choice!!\n\n");

}

printf("PRIORITY QUEUE OPERATIONS\n1: Insert\n2: Delete\n3: Display\n");

printf("\nEnter your choice: ");

scanf("%d",&choice);

}while(choice != -1);

}

-------------------------------------------------------------------------------------------------------------------------------

**priorityheap.h**

struct elementtype{

char name[20];

int id;

int salary;

};

typedef struct elementtype heapdatatype;

struct PQueue{

int capacity;

int size;

heapdatatype \*element;

};

typedef struct PQueue heap;

int isfull(heap \*h){

return h->capacity==h->size;

}

int isempty(heap \*h){

return h->size==0;

}

void display(int size,heap \*h){

printf("-----------------------------------------\n");

for(int i=1;i<size+1;i++)

printf("Name: %s\nId: %d\nSalary: %d\n\n",h->element[i].name,h->element[i].id,h->element[i].salary);

printf("-----------------------------------------\n");

}

heap \*init(int maxelements){

heap \*h;

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

if(h==NULL){

printf("Out of space\n");

return NULL;

}

h->element=(heapdatatype \*)malloc((maxelements+1)\*sizeof(heapdatatype));

if(h->element==NULL){

printf("Out of Space\n");

return NULL;

}

h->capacity=maxelements;

h->size=0;

h->element[0].salary=MAXDATA;

return h;

}

void insert(heapdatatype x,heap \*h){

int i;

if(isfull(h)){

printf("PQueue is full\n");

return;

}

for(i=++h->size;h->element[i/2].salary < x.salary; i/=2)

h->element[i] = h->element[i/2];

h->element[i]=x;

display(h->size,h);

}

heapdatatype deletemin(heap \*h){

int i, child;

heapdatatype maxelement, lastelement;

if(isempty(h)){

printf("PQueue is empty\n");

return h->element[0];

}

maxelement=h->element[1];

lastelement=h->element[h->size--];

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

child=i\*2;

if(child!=h->size && h->element[child+1].salary > h->element[child].salary)

child++;

if(lastelement.salary < h->element[child].salary)

h->element[i]=h->element[child];

else

break;

}

h->element[i]=lastelement;

return maxelement;

}

-------------------------------------------------------------------------------------------------------------------------------**OUTPUT**

Enter the max number of element in the heap: 6

PRIORITY QUEUE OPERATIONS

1: Insert

2: Delete

3: Display

Enter your choice: 1

Enter the employee's name: John

Enter the employee's id: 67

Enter the employee's salary: 55000

-----------------------------------------

Name: John

Id: 67

Salary: 55000

-----------------------------------------

PRIORITY QUEUE OPERATIONS

1: Insert

2: Delete

3: Display

Enter your choice: 1

Enter the employee's name: Peter

Enter the employee's id: 21

Enter the employee's salary: 12000

-----------------------------------------

Name: John

Id: 67

Salary: 55000

Name: Peter

Id: 21

Salary: 12000

-----------------------------------------

PRIORITY QUEUE OPERATIONS

1: Insert

2: Delete

3: Display

Enter your choice: 1

Enter the employee's name: Tyler

Enter the employee's id: 10

Enter the employee's salary: 120000

-----------------------------------------

Name: Tyler

Id: 10

Salary: 120000

Name: Peter

Id: 21

Salary: 12000

Name: John

Id: 67

Salary: 55000

-----------------------------------------

PRIORITY QUEUE OPERATIONS

1: Insert

2: Delete

3: Display

Enter your choice: 1

Enter the employee's name: Jack

Enter the employee's id: 38

Enter the employee's salary: 89

-----------------------------------------

Name: Tyler

Id: 10

Salary: 120000

Name: Peter

Id: 21

Salary: 12000

Name: John

Id: 67

Salary: 55000

Name: Jack

Id: 38

Salary: 89

-----------------------------------------

PRIORITY QUEUE OPERATIONS

1: Insert

2: Delete

3: Display

Enter your choice:

1

Enter the employee's name: Cody

Enter the employee's id: 23

Enter the employee's salary: 89000

-----------------------------------------

Name: Tyler

Id: 10

Salary: 120000

Name: Cody

Id: 23

Salary: 89000

Name: John

Id: 67

Salary: 55000

Name: Jack

Id: 38

Salary: 89

Name: Peter

Id: 21

Salary: 12000

-----------------------------------------

PRIORITY QUEUE OPERATIONS

1: Insert

2: Delete

3: Display

Enter your choice: 1

Enter the employee's name: Marques

Enter the employee's id: 33

Enter the employee's salary: 92000

-----------------------------------------

Name: Tyler

Id: 10

Salary: 120000

Name: Cody

Id: 23

Salary: 89000

Name: Marques

Id: 33

Salary: 92000

Name: Jack

Id: 38

Salary: 89

Name: Peter

Id: 21

Salary: 12000

Name: John

Id: 67

Salary: 55000

-----------------------------------------

PRIORITY QUEUE OPERATIONS

1: Insert

2: Delete

3: Display

Enter your choice: 2

PRIORITY QUEUE OPERATIONS

1: Insert

2: Delete

3: Display

Enter your choice: 3

-----------------------------------------

Name: Marques

Id: 33

Salary: 92000

Name: Cody

Id: 23

Salary: 89000

Name: John

Id: 67

Salary: 55000

Name: Jack

Id: 38

Salary: 89

Name: Peter

Id: 21

Salary: 12000

-----------------------------------------

PRIORITY QUEUE OPERATIONS

1: Insert

2: Delete

3: Display

Enter your choice: 2

PRIORITY QUEUE OPERATIONS

1: Insert

2: Delete

3: Display

Enter your choice: 3

-----------------------------------------

Name: Cody

Id: 23

Salary: 89000

Name: Peter

Id: 21

Salary: 12000

Name: John

Id: 67

Salary: 55000

Name: Jack

Id: 38

Salary: 89

-----------------------------------------

PRIORITY QUEUE OPERATIONS

1: Insert

2: Delete

3: Display

Enter your choice: -1