**Exercise 9: Binary Heap and Application**

**V1:**

**ADT.h:**

#include<stdio.h>

#include<stdlib.h>

struct PQ

{

int ele[100];

int size;

int ctptr;

};

void init(struct PQ \*,int);

void display(struct PQ \*);

void insert(struct PQ \*,int);

int del(struct PQ \*);

**impl.h:**

#include "adt.h"

void init(struct PQ \*P,int s)

{

P->size=s;

P->ctptr=0;

P->ele[0]=-10;

}

void display(struct PQ \*P)

{

printf("PQ:\n ");

for(int i=0;i<=P->ctptr;i++)

{

printf("%d ",P->ele[i]);

}

printf("\n");

}

void insert(struct PQ \*P,int x)

{

P->ctptr++;

P->ele[P->ctptr]=x;

for(int i=P->ctptr;P->ele[i/2]>x;i=i/2)

{

P->ele[i]=P->ele[i/2];

P->ele[i/2]=x;

}

}

int del(struct PQ \*P)

{

int min,tmp,child,i;

min=P->ele[1];

P->ele[1]=P->ele[P->ctptr--];

for(int i=1;2\*i<=P->ctptr;i=child)

{

child=2\*i;

if(child+1<=P->ctptr&&P->ele[child]>P->ele[child+1])

child=child+1;

if(P->ele[child]<P->ele[i])

{

tmp=P->ele[i];

P->ele[i]=P->ele[child];

P->ele[child]=tmp;

}

}

return min;

}

**appl.c:**

#include "impl.h"

int main()

{

struct PQ \*p=(struct PQ \*)malloc(sizeof(struct PQ));

init(p,20);

/\*insert(P,20);

insert(P,1);

insert(P,10);

insert(P,15);

insert(P,19);\*/

/\*insert(p,14);

insert(p,16);

insert(p,22);

insert(p,11);

insert(p,9);

insert(p,18);

insert(p,10);

insert(p,7);

insert(p,4);

insert(p,1);\*/

printf("\nEnter the number of elements: ");

int n,ele;

scanf("%d",&n);

for(int i=0;i<n;i++)

{

printf("Enter the element: ");

scanf("%d",&ele);

insert(p,ele);

}

/\*insert(p,'A');

insert(p,'K');

insert(p,'R');

insert(p,'T');

insert(p,'L');

insert(p,'P');

insert(p,'Y');\*/

/\*insert(p,15000);

insert(p,12000);

insert(p,4000);

insert(p,3500);

insert(p,4600);

insert(p,6000);

insert(p,8600);\*/

display(p);

printf("\nDeletion:");

/\*int a=del(p);

int b=del(p);

display(p);

printf("a= %d,b= %d\n",a,b);\*/

int b;

for(int i=0;i<2;i++)

{

b=del(p);

display(p);

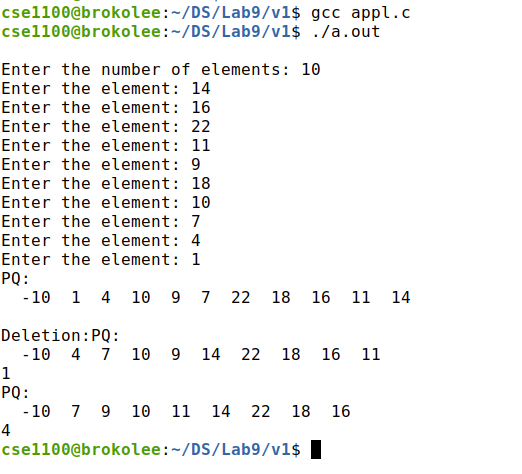
printf("%d\n",b);

}

return 0;

}

**Sample I/O:**

****

**V2:**

**ADT.h:**

#include<stdio.h>

#include<stdlib.h>

struct emp

{

char id;

int sal;

};

struct PQ

{

struct emp E[100];

int size;

int ctptr;

};

void init(struct PQ \*,int);

void display(struct PQ \*);

void insert(struct PQ \*,struct emp);

struct emp del(struct PQ \*);

**impl.h:**

#include "adt.h"

void init(struct PQ \*P,int s)

{

P->size=s;

P->ctptr=0;

P->E[0].sal=-10;

P->E[0].id='\0';

}

void display(struct PQ \*P)

{

printf("\nID\tSalary\n");

for(int i=1;i<=P->ctptr;i++)

{

printf("%c\t%d\n",P->E[i].id,P->E[i].sal);

}

printf("\n");

}

void insert(struct PQ \*P,struct emp E)

{

P->ctptr++;

P->E[P->ctptr]=E;

//P->E[P->ctptr].id=E.id;

for(int i=P->ctptr;P->E[i/2].sal>E.sal;i=i/2)

{

P->E[i]=P->E[i/2];

P->E[i/2]=E;

//P->E[i]=P->E[i/2];

//P->E[i/2].id=E.id;

}

}

struct emp del(struct PQ \*P)

{

/\*int child,i,t;

char t2;

struct emp \*E=(struct emp\*)malloc(sizeof(struct emp));

E->sal=P->E[1].sal;

E->id=P->E[1].id;

P->E[1].sal=P->E[P->ctptr].sal;

P->E[1].id=P->E[P->ctptr--].id;

for(i=i;2\*i<=P->ctptr;i=child)

{

child=2\*i;

if(P->E[child].sal<P->E[i].sal || P->E[child+1].sal<P->E[i].sal)

{

if(P->E[child+1].sal<P->E[child].sal)

child++;

}

if(P->E[i].sal>P->E[child].sal)

{

t=P->E[i].sal;

t2=P->E[i].id;

P->E[i].sal=P->E[child].sal;

P->E[i].id=P->E[child].id;

P->E[child].sal=t;

P->E[child].id=t2;

}

}

//printf("\nhere\n");

return E;\*/

int child;

struct emp tmp,ele=P->E[1];

P->E[1]=P->E[P->ctptr--];

for(int i=1;2\*i<=P->ctptr;i\*=2)

{

child=2\*i;

if(P->E[2\*i].sal>P->E[2\*i+1].sal)

child=2\*i+1;

if(P->E[child].sal<P->E[i].sal)

{

tmp=P->E[i];

P->E[i]=P->E[child];

P->E[child]=tmp;

}

}

return ele;

}

**appl.c:**

#include "impl.h"

int main()

{

struct PQ \*p=(struct PQ \*)malloc(sizeof(struct PQ));

init(p,20);

/\*insert(P,20);

insert(P,1);

insert(P,10);

insert(P,15);

insert(P,19);\*/

struct emp E,E2;//=(struct emp \*)malloc(sizeof(struct emp));

/\*printf("Heap:\n");

E.sal=15000;

E.id='A';

insert(p,E);

E.sal=12000;

E.id='K';

insert(p,E);

E.sal=4000;

E.id='R';

insert(p,E);

E.sal=3500;

E.id='T';

insert(p,E);

E.sal=4600;

E.id='L';

insert(p,E);

E.sal=6000;

E.id='P';

insert(p,E);

E.sal=8600;

E.id='Y';

insert(p,E);\*/

printf("\nEnter the number of Employees: ");

int n;

char id;

int sal;

scanf("%d",&n);

for(int i=0;i<n;i++)

{

printf("Enter the emp id: ");

scanf(" %c",&id);

printf("Enter the salary: ");

scanf(" %d",&sal);

E.sal=sal;

E.id=id;

insert(p,E);

}

display(p);

printf("Deletion order:\n");

printf("\n");

//E2=del(p);

//int \*B=del(p);

//display(p);

//printf("%c\t%d\n",E2->id,E2->sal);

for(int i=0;i<7;i++)

{

E2=del(p);

printf("%c\t%d\n",E2.id,E2.sal);

}

return 0;

}

**Sample I/O:**

