**PROGRAM 10 :**

**Write a C program to implement priority queue to insert, delete and display the elements.**

**PROGRAM :**

#include<stdio.h>

#include<stdlib.h>

#define size 100

void heapify(int a[], int n)

{

int key, k, heap, j;

for( k=n/2; k>=1; k--)

{

key = a[k];

heap =0;

while( !heap && (2\*k)<=n)

{

j=2\*k ;

if( j < n && a[j+1] > a[j])

j= j+1;

if( key < a[j] )

{

a[k]=a[j];

k=j;

}

else

heap =1;

}

a[k] = key;

}

}

int main()

{

int i, a[size], n,ele,choice;

printf("Enter n\n");

scanf("%d", &n);

printf("Enter Data\n");

for( i=1; i<=n; i++ )

scanf("%d", &a[i]);

heapify(a, n);

while(1)

{

printf("\nEnter the choice \n1 : insert \n2 : delete\n3 : display\n4 : exit\n");

scanf("%d",&choice);

switch(choice)

{

case 1: if( n <size)

{

printf("\nEnter the key to be inerted :");

scanf("%d",&ele);

n++;

a[n] =ele;

heapify(a, n);

}

else

printf("PRIORITY QUEUE FULL !!");

break;

case 2: if( n > 0)

{

ele=a[1];

a[1]=a[n];

n--;

heapify(a, n);

printf("\nDeleted element is %d :",ele);

}

else

printf(" PRIORITY QUEUE EMPTY !");

break;

case 3: printf(" Heap is :");

for( i=1; i<=n; i++ )

printf("%d ",a[i]);

break;

default:exit(0);

}

}

}

**OUTPUTS :**

**OUTPUT 1 :**

Enter n

5

Enter Data

1

2

3

4

5

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

3

Heap is :5 4 3 1 2

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

2

Deleted element is 5 :

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

2

Deleted element is 4 :

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

3

Heap is :3 2 1

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

1

Enter the key to be inerted :6

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

1

Enter the key to be inerted :7

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

3

Heap is :7 6 1 2 3

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

4

**OUTPUT 2 :**

Enter n

5

Enter Data

1

2

3

4

5

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

2

Deleted element is 5 :

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

1

Enter the key to be inerted :10

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

2

Deleted element is 10 :

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

3

Heap is :4 2 3 1

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

4

**OUTPUT 3 :**

Enter n

1

Enter Data

10

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

2

Deleted element is 10 :

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

2

PRIORITY QUEUE EMPTY !

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

1

Enter the key to be inerted :20

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

1

Enter the key to be inerted :30

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

1

Enter the key to be inerted :40

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

1

Enter the key to be inerted :50

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

3

Heap is :50 40 30 20

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

2

Deleted element is 50 :

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

3

Heap is :40 20 30

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

2

Deleted element is 40 :

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

3

Heap is :30 20

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

2

Deleted element is 30 :

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

3

Heap is :20

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

2

Deleted element is 20 :

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

3

Heap is :

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

2

PRIORITY QUEUE EMPTY !

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

3

Heap is :

Enter the choice

1 : insert

2 : delete

3 : display

4 : exit

4