Ex.No: 11   Implementation of Priority Queues using Binary Heap

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define MAX 20
void main()
{
 int choice,top=1,a[20],item,i,min;
 char str='c';
 void insert(int[],int,int);
 void DeleteMin(int[],int);
 int FindMin(int [],int);
 void display(int [],int);
 clrscr();
 printf("Priority Queue Implementation(Binary Heaps)");
 printf("\n--------------------------------------------------\n");
 while(str=='c')
 {
  printf("\n 1. Insert\n");
  printf(" 2. DeleteMin\n");
  printf(" 3. FindMin\n");
  printf(" 4. Display\n");
  printf(" 5. Exit\n");
  printf(" Enter your Choice :");
  scanf("%d",&choice);

  switch(choice)
  {
   case 1:
    if(top<=MAX)
    {
     printf("\n Enter the element :");
     scanf("%d",&item);
     insert(a,top,item);
     top++;
     display(a,top);
    }
    else
     printf("Priority Queue is Full");
    break;
   case 2:
    if(top>0)
    {
     DeleteMin(a,top);
     top--;
     display(a,top);
    }
    else
     printf(" Priority Queue is Empty");
    break;
   case 3:
    if(top>0)
    {
     min=FindMin(a,top);
     printf(" Minimum Element is:%d",min);
    }
    break;
   case 4:
    printf(" Elements in the priority queue are:");
    display(a,top);
    break;
   case 5:
    exit(0);
  } //switch
 printf(" \n Press c to continue:\n");
 str=getch();
 } //while
}  //main

void insert(int a[20],int top,int item)
{
int i,temp;
a[top]=item;
i=top;
while((i/2)>0)
{
 if(a[i]<a[i/2])
 {
  temp=a[i];
  a[i]=a[i/2];
  a[i/2]=temp;
 } //if
 i=i/2;
} //while
} //insert
```

```
void DeleteMin(int a[20],int top)
{
int i,j,temp;
a[1]=a[top-1];
top--;
i=1;
while((i*2)+1<top)
{
 if(a[2*i]>a[(2*i)+1])
  j=(2*i)+1;
 else
  j=2*i;
 if(a[i]>a[j])
 {
  temp=a[i];
  a[i]=a[j];
  a[j]=temp;
  i=j;
  }//if
 }//while
 if(2*i<top)
  if(a[i]>a[2*i])
   {
  temp=a[i];
  a[i]=a[2*i];
  a[2*i]=temp;
   }
 }//process
int FindMin(int a[20],int top)
{
 if(top<1)
 {
  printf( "Priority Queue is Empty" );
  return 0;
 }
 else
  return a[1];
}

void display(int a[20],int top)
{
int i;
for(i=1;i<top;i++)
 printf(" %d   ",a[i]);
}
```

OUTPUT

```
Priority Queue Implementation(Binary Heaps)
-----------------------------------------------------

 1. Insert
 2. DeleteMin
 3. FindMin
 4. Display
 5. Exit
 Enter your Choice : 1

 Enter the element : 7
 7
 Press c to continue:

 1. Insert
 2. DeleteMin
 3. FindMin
 4. Display
 5. Exit
 Enter your Choice : 1

 Enter the element : 3
 3    7
 Press c to continue:

 1. Insert
 2. DeleteMin
 3. FindMin
 4. Display
 5. Exit
 Enter your Choice : 1

 Enter the element : 5
 3    7    5
 Press c to continue:

 1. Insert
 2. DeleteMin
 3. FindMin
 4. Display
 5. Exit
 Enter your Choice : 1

 Enter the element : 4
```

```
3    4    5    7
Press c to continue:

1. Insert
2. DeleteMin
3. FindMin
4. Display
5. Exit
Enter your Choice : 1

Enter the element : 8
3    4    5    7    8
Press c to continue:

1. Insert
2. DeleteMin
3. FindMin
4. Display
5. Exit
Enter your Choice : 1

Enter the element : 6
3    4    5    7    8    6
Press c to continue:

1. Insert
2. DeleteMin
3. FindMin
4. Display
5. Exit
Enter your Choice : 1

Enter the element : 2
2    4    3    7    8    6    5
Press c to continue:

1. Insert
2. DeleteMin
3. FindMin
4. Display
5. Exit
Enter your Choice : 1

 Enter the element : 9
 2    4    3    7    8    6    5    9
 Press c to continue:
```

```
 1. Insert
 2. DeleteMin
 3. FindMin
 4. Display
 5. Exit
 Enter your Choice : 1

 Enter the element : 1
1    2    3    4    8    6    5    9    7
 Press c to continue:

 1. Insert
 2. DeleteMin
 3. FindMin
 4. Display
 5. Exit
 Enter your Choice : 4
 Elements in the priority queue are: 1    2    3    4    8    6    5    9    7

 Press c to continue:

 1. Insert
 2. DeleteMin
 3. FindMin
 4. Display
 5. Exit
 Enter your Choice : 2
2    4    3    7    8    6    5    9
Press c to continue:

1. Insert
2. DeleteMin
3. FindMin
4. Display
5. Exit
Enter your Choice : 2
3    4    5    7    8    6    9
Press c to continue:

1. Insert
2. DeleteMin
3. FindMin
4. Display
5. Exit
Enter your Choice : 3
Minimum Element is: 3
```

```
Press c to continue:

1. Insert
2. DeleteMin
3. FindMin
4. Display
5. Exit
Enter your Choice : 4
Elements in the priority queue are: 3    4    5    7    8    6    9
Press c to continue:

1. Insert
2. DeleteMin
3. FindMin
4. Display
5. Exit
Enter your Choice : 5
```