Name-Arya Dubey

Registration Number-20BCE0908

Faculty: Professor Gopinath M.P.

IMPLEMENTATION OF HEAP

1)Min heap

```
#include<iostream.h>
#include<stdlib.h>
#define maxsize 100

class heap
{
        int curr_size;
             int a[maxsize];
        public:
                void initialize();
                void buildheap();
                void percolatedown(int hole);
                void insert(int x);
                void delmin(int &min);
                void display();
                 void findmin();
};

void heap :: initialize()
{
           int i;
```

```
        cout << "\n Enter the size of the array :";
        cin >> curr_size;
        cout << "\nEnter " << curr_size << " elemets\n";
        for(i=1;i<=curr_size;i++)
        cin >> a[i];
        buildheap();
}

void heap :: buildheap()
{
        int i;
        for(i = curr_size/2;i>0;i--)
        percolatedown(i);
}

void heap :: insert(int x)
{
    if(curr_size == maxsize)
            cout <<"\n Array is full\n";
    else
    {
                    int hole = ++curr_size;
                    for(;hole > 1 && x < a[hole/2] ; hole = hole / 2)
            a[hole] = a[hole/2];
                    a[hole] = x;
    }
}

void heap :: delmin(int &min)
{
    if(curr_size == 0)
    cout << "\narray is empty ";
    else
    {
```

```
                        min = a[1];
                        a[1] = a[curr_size--];
                         percolatedown(1);
            }
}

void heap :: percolatedown(int hole)
{
      int child;
      int tmp = a[hole];
      for(; hole * 2 <= curr_size; hole=child)
      {
            child=hole * 2;
            if(child != curr_size && a[child+1] < a[child])
                  child++;
            if(a[child] < tmp)
                  a[hole] = a[child];
            else
               break;
      }
      a[hole] = tmp;
}

void heap :: display()
{
            int i;
            cout<<"\n\n\tIndex\telement\n";
            for(i=1;i<=curr_size;i++)
            cout <<"\n\n\t"<<i<<"\t"<<a[i]<<'\n';
}



void heap::findmin()
```

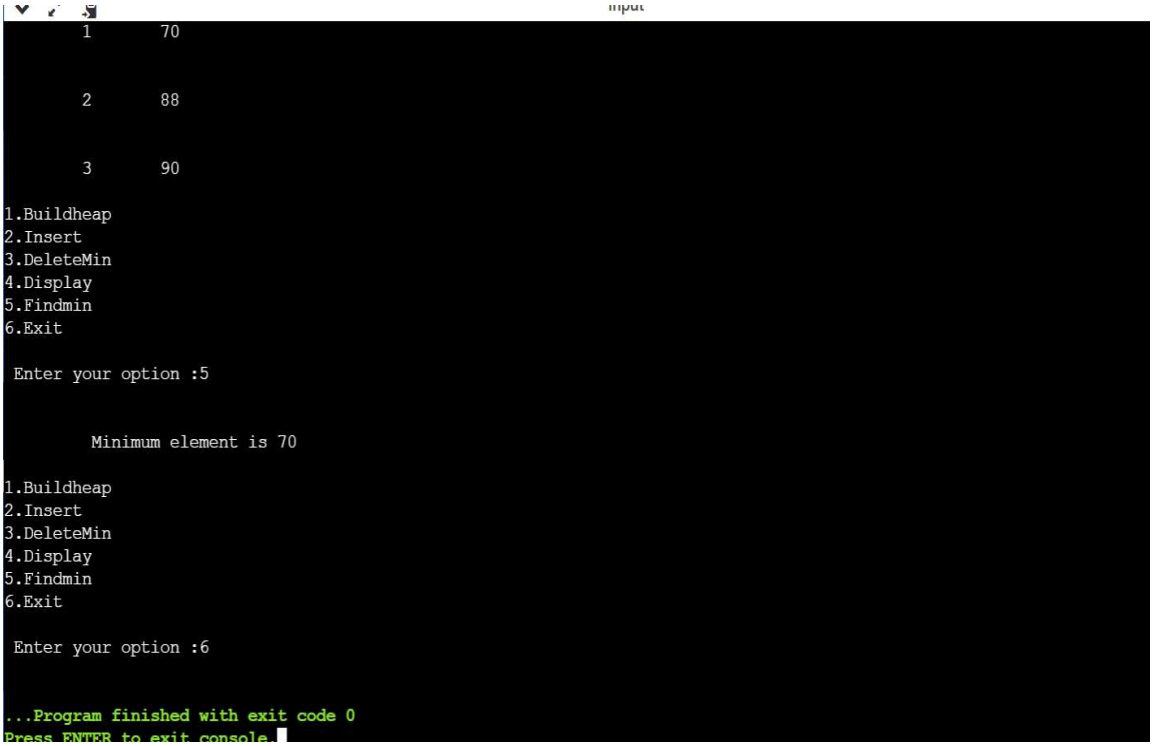
```
{
 cout<<"\n\n\t Minimum element is "<<a[1]<<endl;
}
int main()
{
      int x,ch,min=0;
      heap h;
      system("clear");
      cout<<  "\n\n\tMin Heap \n";
      do
      {
    cout<<"\n1.Buildheap\n2.Insert\n3.DeleteMin\n4.Display\n5.Findmin\n6.Exit\n";
            cout << "\n Enter your option :";
            cin >> ch;
            switch(ch)
                  {
                        case 1: h.initialize();   break;
                        case 2: cout << "\nEnter the data to be inserted\n";
                        cin >> x;
                        h.insert(x);
                        break;
                        case 3: h.delmin(min);
                        cout <<"\nMinimum element "<<min<<" is deleted\n";
                        break;
                        case 4: h.display(); break;
                        case 5: h.findmin(); break;
                        case 6: exit(0);
                        default:cout<<"\n opt the right choice \n";
          }
        }
          while(ch != 6);
          return 0;
}
```

```
        Min Heap

1.Buildheap
2.Insert
3.DeleteMin
4.Display
5.Findmin
6.Exit

 Enter your option :1

 Enter the size of the array :3

Enter 3 elemets
50
70
90

1.Buildheap
2.Insert
3.DeleteMin
4.Display
5.Findmin
6.Exit

 Enter your option :4


        Index   element
```

```
       Index   element


       1       50


       2       70


       3       90

.Buildheap
.Insert
.DeleteMin
.Display
.Findmin
.Exit

 Enter your option :2

nter the data to be inserted
8

.Buildheap
.Insert
.DeleteMin
.Display
.Findmin
.Exit

 Enter your option :4
```

```
5.Findmin
6.Exit

 Enter your option :4


       Index   element


       1       50


       2       70


       3       90


       4       88

1.Buildheap
2.Insert
3.DeleteMin
4.Display
5.Findmin
6.Exit

 Enter your option :3

Minimum element 50 is deleted

1.Buildheap
```

```
Minimum element 50 is deleted

1.Buildheap
2.Insert
3.DeleteMin
4.Display
5.Findmin
6.Exit

 Enter your option :4


        Index    element


        1        70


        2        88


        3        90

1.Buildheap
2.Insert
3.DeleteMin
4.Display
5.Findmin
6.Exit

 Enter your option :5
```



---

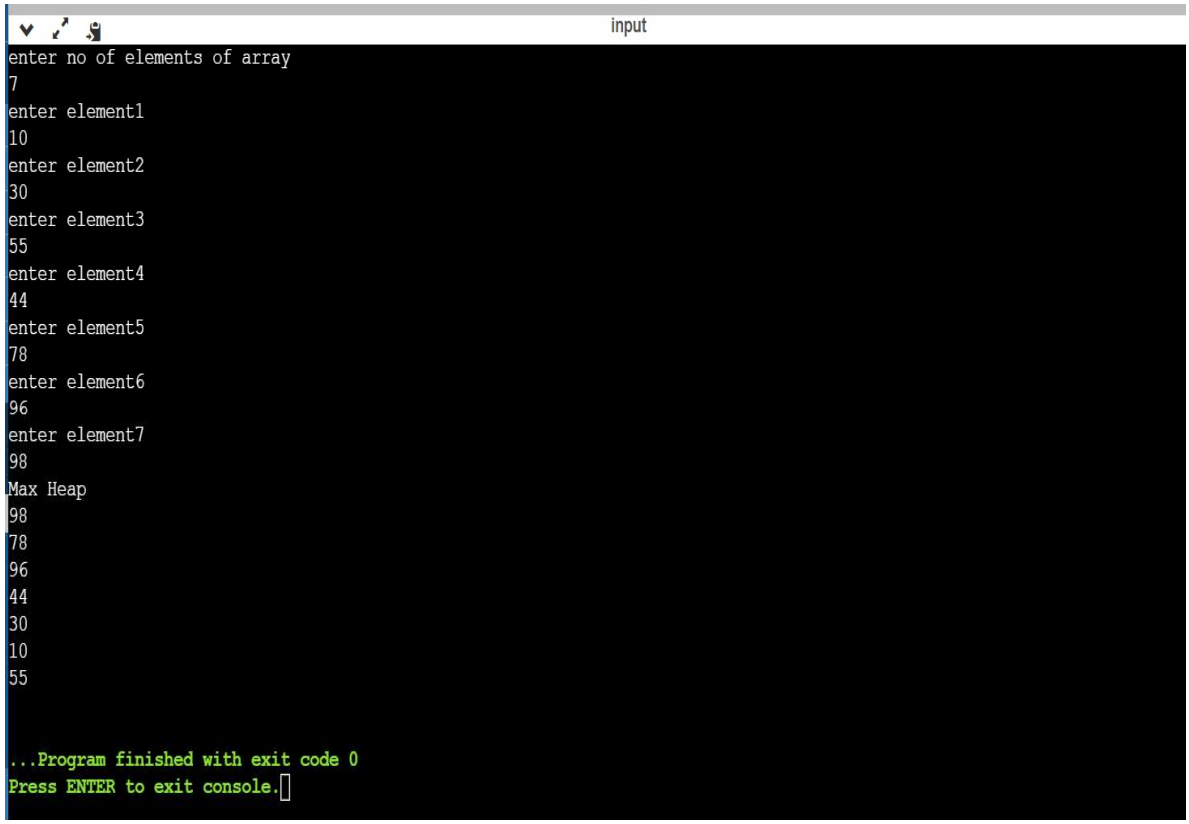
2)Max Heap

```
#include <iostream>
#include <conio.h>
using namespace std;
void max_heapify(int *a, int i, int n)
{
    int j, temp;
    temp = a[i];
    j = 2 * i;
    while (j <= n)
```

```
    {
        if (j < n && a[j+1] > a[j])
            j = j + 1;
        if (temp > a[j])
            break;
        else if (temp <= a[j])
        {
            a[j / 2] = a[j];
            j = 2 * j;
        }
    }
    a[j/2] = temp;
    return;
}
void build_maxheap(int *a,int n)
{
    int i;
    for(i = n/2; i >= 1; i--)
    {
        max_heapify(a,i,n);
```

```
    }
}
int main()
{
    int n, i, x;
    cout<<"enter no of elements of array\n";
    cin>>n;
    int a[20];
    for (i = 1; i <= n; i++)
    {
        cout<<"enter element"<<(i)<<endl;
        cin>>a[i];
    }
    build_maxheap(a,n);
    cout<<"Max Heap\n";
    for (i = 1; i <= n; i++)
    {
        cout<<a[i]<<endl;
    }
    getch();
```

}


input
enter no of elements of array
7
enter element1
10
enter element2
30
enter element3
55
enter element4
44
enter element5
78
enter element6
96
enter element7
98
Max Heap
98
78
96
44
30
10
55
...Program finished with exit code 0
Press ENTER to exit console.