
Assignment Name: Implementation of Max Heap Tree using Insert

----

```
#include<iostream.h>
#include<conio.h>
class heap
{
      int n,a[10],q,i,j,k,key;
public:
      void get();
      void create();
      void display();
};
void heap::get()
{
      cout<<"\nEnter Range:";
      cin>>n;
      cout<<"\nEnter the element:";
      for(i=1;i<=n;i++)
       cin>>a[i];
}
void heap::create()
{
      for(q=2;q<=n;q++)
      {
       i=q;
       key=a[q];
       j=i/2;
       while(i>1 && key>a[j]) //change
       {
            a[i]=a[j];
            i=j;
            j=i/2;

            if(j<1)
             j=1;
       }
      a[i]=key;
      }
}
void heap::display()
{
      cout<<"\nHeap Tree:";
      for(i=1;i<=n;i++)
       cout<<a[i]<<"\t";
}
void main()
{
      clrscr();
      heap h;
      h.get();
      h.create();
      h.display();
      getch();
}
*/ Output */
Enter Range:7
Enter the element:80 45 70 40 35 50 90
Heap Tree:90     45      80      40      35      50      70
```

Assignment Name: Implementation of Min Heap Tree using Insert

----

```
#include<iostream.h>
#include<conio.h>
class heap
{
     int n,a[10],q,i,j,k,key;
public:
     void get();
     void create();
     void display();
};
void heap::get()
{
     cout<<"\nEnter Range:";
     cin>>n;
     cout<<"\nEnter the element:";
     for(i=1;i<=n;i++)
      cin>>a[i];
}
void heap::create()
{
     for(q=2;q<=n;q++)
     {
      i=q;
      key=a[q];
      j=i/2;
      while(i>1 && key<a[j]) //change
      {
          a[i]=a[j];
          i=j;
          j=i/2;

          if(j<1)
           j=1;
      }
     a[i]=key;
     }
}
void heap::display()
{
     cout<<"\nHeap Tree";
     for(i=1;i<=n;i++)
      cout<<a[i]<<"\t";
}
void main()
{
     clrscr();
     heap h;
     h.get();
     h.create();
     h.display();
     getch();
}
*/ Output */
Enter Range:7
Enter the element:80 45 70 40 35 50 90
Heap Tree35     40      50      80      45      70      90
```

Assignment Name: Implementation of Max heap using Heapify/Adjust

----

```
#include<iostream.h>
#include<conio.h>
```

```
class heap
{
  int i,j,item,a[10],n;
  public:
   void get();
   void show();
   void adjust(int [],int i,int j);
   void heapify(int [],int);
 };
void heap::get()
{
 cout<<"enter the size of array";
 cin>>n;
 for(i=1;i<=n;i++)
 cin>>a[i];
 heapify(a,n);
}
void heap::show()
{
  cout<<"\nThe element is=>\n";
  for(i=1;i<=n;i++)
  cout<<a[i]<<"\t";
}
void heap::adjust(int a[],int i,int n)
{
  j=2*i;
  item=a[i];
  while(j<=n)
  {
    if((j<n)&&(a[j]<a[j+1]))
    j++;
    if(item>=a[j])
    break;
    a[j/2]=a[j];
    j=2*j;
   }
   a[j/2]=item;
}
void heap::heapify(int a[],int n)
{
  for(i=n/2;i>=1;i--)
  adjust(a,i,n);
}
void main()
{
 clrscr();
 heap h;
 h.get();
 h.show();
 getch();
}




/*
output==>

enter the size of array

7
```

```
element are=>

10      3       335     33      355     217     536

ele after max heap=>

536     355     335     33      3       217     10

*/
```

Assignment Name: Implementation of Min Heap using Heapify / Adjust

----

```
#include<iostream.h>
#include<conio.h>

class heap
{
   int i,j,item, a[1000],n;
   public:
   void get();
   void show();
   void adjust(int [],int i,int j);
   void heapify(int [],int);
 };
void heap::get()
{
 cout<<"enter the size of array";
 cin>>n;
 for(i=1;i<=n;i++)
 cin>>a[i];
 heapify(a,n);
}
void heap::show()
{
  cout<<"\nthe element is=>\n";
  for(i=1;i<=n;i++)
  cout<<a[i]<<"\t";
}
void heap::adjust(int a[],int i,int n)
{
  j=2*i;
  item=a[i];
  while(j<=n)
  {
    if((j<n)&&(a[j]>a[j+1]))
    j++;
    if(item<=a[j])
    break;
    a[j/2]=a[j];
    j=2*j;
   }
   a[j/2]=item;
}
void heap::heapify(int a[],int n)
{
  for(i=n/2;i>=1;i--)
  adjust(a,i,n);
}
void main()
{
 clrscr();
 heap h;
 h.get();
 h.show();
 getch();
}




/*
output==>

enter the size of array
7

element are=>
10      3       335     33      355     217     536

element are=>
3       10      217     33      355     335     536

*/
```