```
#include <stdio.h>
int k=0, size;
void insertheap(int ar[], int val);
void heapsort(int ar[]);
void main()
{
   int i,el,ch=0,j=1;
   printf("Enter max size : ");
   scanf("%d", &size);
   int ar[size];
   while(ch!=4)
   {
      printf("\n\nMENU\n1.Insertion\n2.Display\n3.Heap 
Sort\n4.Exit\nEnter your choice : ");
      scanf("%d",&ch);
      switch(ch)
      {
         case 1:
            if(j<=size)
            {
               printf("Enter an element : ");
               scanf("%d", &el);
               insertheap(ar,el);
               j++;
            }
            else
               printf("Insertion not possible");
            break;
         case 2:
            printf("Heap Tree \n");
            for(int i=1;i<=k;i++)
               printf("%d\t", ar[i]);
            printf("\n");
            break;
         case 3:
            heapsort(ar);
            printf("Sorting Done");
            break;
         case 4:
            printf("Exitted");
            break;
         default:
            printf("Wrong Choice");

      }
   }
}

void insertheap(int ar[], int val)
{
     int i,parent,temp;;
     if(k>=size)
     printf("Insertion not possible\n");
     else
     {
          k++;
          ar[k]=val;
          i=k;
          while(i > 1)
          {
               parent = i/2;
               if(ar[parent]<=ar[i])
               {
                    temp=ar[parent];
                    ar[parent]=ar[i];
                    ar[i]=temp;
               }
               i = parent;
          }
     }
}

void heapsort(int ar[])
{
     int i,x,y,temp,j,flag;
     i=k;
     while(i > 1)
     {
          temp=ar[i];
          ar[i]=ar[1];
          ar[1]=temp;
          i--;
          j = 1;
          flag = 0;
          while(j<i && flag==0)
          {
               int lc = 2*j;
               int rc = 2*j+1;
               if(lc < i)
                    x = ar[lc];
               else
                    x = -1;
               if(rc < i)
                    y = ar[rc];
               else
                    y = -1;
               if(ar[j] > x && ar[j] > y)
                    flag = 1;
               else
               {
                    if(x > y && ar[j] < x)
                    {
                         temp=ar[j];
                         ar[j]=ar[lc];
                         ar[lc]=temp;
                         j = lc;
                    }
                    else if(y > x && ar[j] < y)
                    {
                         temp=ar[j];
                         ar[j]=ar[rc];
                         ar[rc]=temp;
                         j = rc;
                    }
               }
          }
     }
}
/*
OUTPUT:
Enter max size : 5


MENU
1.Insertion
2.Display
3.Heap Sort
4.Exit
Enter your choice : 1
Enter an element : 23


MENU
1.Insertion
2.Display
```

```
3.Heap Sort
4.Exit
Enter your choice : 1
Enter an element : 34


MENU
1.Insertion
2.Display
3.Heap Sort
4.Exit
Enter your choice : 1
Enter an element : 45


MENU
1.Insertion
2.Display
3.Heap Sort
4.Exit
Enter your choice : 1
Enter an element : 67


MENU
1.Insertion
2.Display
3.Heap Sort
4.Exit
Enter your choice : 1
Enter an element : 78


MENU
1.Insertion
2.Display
3.Heap Sort
4.Exit
Enter your choice : 2
Heap Tree
78 67 34 23 45


MENU
1.Insertion
```

```
2.Display
3.Heap Sort
4.Exit
Enter your choice : 3
Sorting Done

MENU
1.Insertion
2.Display
3.Heap Sort
4.Exit
Enter your choice : 2
Heap Tree
23 34 45 67 78

*/
```