```
#include<stdio.h>


void heaptree(int a[], int n,int e)
{
   int i,t;
   i = n;
   a[i] = e;
   while(i != 0)
   {
      if(a[i] < a[(i-1)/2])
      {
         break;
      }
      else
      {
         t = a[i];
         a[i] = a[(i-1)/2];
         a[(i-1)/2] = t;
      }
      i = (i - 1)/2;
   }
}
void disparr(int a[],int n)
{
   int i;
   for(i = 0;i <= n;i++)
   {
      printf("%d\t",a[i]);
   }
}
```

```
void heapsort(int a[],int n)
{
   int i,j,t,lc,rc;
   j = n;

   while(j > 0)
   {
      i = 0;
      t = a[i];
      a[i] = a[j];
      a[j] = t;
      j--;
      while(i < j )
      {
         lc = (2 * i) + 1;
         rc = (2 * i) + 2;
         if(lc <= j && rc <= j)
         {
            if((a[i] > a[lc]) && (a[i] > a[rc]) )
            {
               break;
            }
            else if(a[lc] > a[rc])
            {
               t = a[i];
               a[i] = a[lc];
               a[lc] = t;
               i=lc;
            }
            else
            {
```

```
                t = a[i];
                a[i] = a[rc];
                a[rc] = t;
                i = rc;
            }
        }
        else if(a[lc] > a[i])
        {
                t = a[i];
                a[i] = a[lc];
                a[lc] = t;
                i=lc;
        }
        else
        {
            break;
        }
      }
   }
}

int main()
{
   int a[5];
   heaptree(a,0,10);
   heaptree(a,1,12);
   heaptree(a,2,15);
   heaptree(a,3,5);
   heaptree(a,4,20);
   disparr(a,4);
   heapsort(a,4);
```

```
    disparr(a,4);
    return 0;
}
```