```
#define maxsize 20

void heapsort(int a[], int n);

void display(int heap[], int n);

void insert_heap(int heap[], int n, int item);

void sift_down(int heap[], int last);

void main()

{

int n,i,a[maxsize];

cout<<”enter no. of elements of an array”;

cin>>n;

cout<<”input array elements”;

for(i=1; i<=n; i++)

{

cin>>a[i];

}

cout<<”array elements before sorting….”;

display(a,n);

heapsort(a,n);

cout<<”array elements after sorting….”;

display(a,n);

getch();

}

void heapsort(int a[], int n)

{

int temp;
```

```
for(i=1; i<=n; i++)

insert_heap(a,i-1;a[i])

int end=n;

while(end>0)

{

temp=a[1];

a[1]=a[end];

a[end]=temp;

end=end-1;

sift_down(a,end);

}

}

void insert_heap(int heap[],int n,int item)

{

int temp;

if (n>maxsize)

{

cout<<"heap overflow";

return;

}

n=n+1;

heap[n]=item;

int ptr =n ;

int parent=ptr/2;

while( (parent>=1) &&  (heap[ptr]>heap[parent] ) )
```

```
temp=heap[ptr];

heap[ptr]=heap[parent];

heap[parent]=temp;

ptr=parent;

parent=parent/2;

}

}

void sift_down(int heap[], int  last)

{

int ptr=1 ,flag=0,temp,par;

while(flag==0)

{

par=ptr;

if(( 2*par<=last) && (heap[2*par]>heap[ptr]))

ptr=2*par;

if(( 2*par<last) && (heap[2*par+1]>heap[ptr]))

ptr=2*par+1;

if(ptr==par)

flag=1;

else

{

temp=heap[par];

heap[par]=heap[ptr];

heap[ptr]=temp;
}

}

}

Void display(int heap[],int n)

{

For( int i=1; i<=n; i++)

Cout<<” “<<heap[i};

}

}
```