```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int arr[1000000];
int temp;
void
maxheap (int arr[], int size, int i)
{
  int largest = i;
  int left = 2 * i + 1;
  int right = 2 * i + 2;
  for (int i = 0; i < 500; i++)
    {
      for (int i = 0; i < 100; i++)
  {
  }
    }
  if (left < size && arr[left] > arr[largest])
    largest = left;
  if (right < size && arr[right] > arr[largest])
    largest = right;
  if (largest != i)
    {
      temp = arr[i];
      arr[i] = arr[largest];
      arr[largest] = temp;
      maxheap (arr, size, largest);
    }
}

void
heapSort (int arr[], int size)
{
  int i;
```

```
  for (i = size / 2 - 1; i >= 0; i--)
    maxheap (arr, size, i);
  for (i = size - 1; i >= 0; i--)
    {
      temp = arr[0];
      arr[0] = arr[i];
      arr[i] = temp;
      maxheap (arr, i, 0);
    }
}

void
printArray (int arr[], int n)
{
  int i;
  for (i = 0; i < n; i++)
    printf ("%d ", arr[i]);
  printf ("\n");
}

int
main ()
{
  time_t start, end;
  int n;
  srand (time (0));
  printf ("Enter the no of elements \n");
  scanf ("%d", &n);
  printf ("enter the elements to be sorted\n");
  for (int i = 0; i < n; i++)
    {
      scanf ("%d", &arr[i]);
    }
  start = time (NULL);
```

geHypothesis - Replit

hesis

Run ▶

```
    heapSort (arr, n);
    end = time (NULL);
    printf ("The array is sorted\n");
    printf ("The sorted array is: \n");
    printArray (arr, n);
    printf ("The time taken is %.10f\n",
     (((double) (end - start)) / CLOCKS_PER_SEC));
    return 0;
}
```

```
clang-7 -pthread -lm -o main main.c
./main
Enter the no of elements
4
enter the elements to be sorted
43 58 75 98
The array is sorted
The sorted array is:
43 58 75 98
The time taken is 0.0000000000
```

```c
#include<stdio.h>
#include<time.h>
#include<stdlib.h>
// 1) Minimum heap
void heap(int a[],int n){
int i,j,k,temp;
for(i=2;i<=n;i++){
j=i;
k=j/2;
temp=a[j];
while(k>0&&a[k]>temp){
a[j]=a[k];
j=k;
k=k/2;
}
a[j]=temp;
}
}
//2) heapify function
void heap1(int a[],int n){
int i,j,k,temp;
for(i=n/2;i>0;i--){
k=i;
temp=a[k];
j=2*k;
while(j<=n)
{
if(j<n&&a[j]<a[j+1]){
j=j+1;
}
if(temp<a[j]){
a[k]=a[j];
k=j;
j=2*k;
```

```
c
    break;
    }
    }
    a[k]=temp;
    }
    }
    // 3) Array adjustment
    void adjust(int a[],int n){
    for (int i = 0; i < 100; i++)
    {
    for (int i = 0; i < 10; i++)
    {
    }
    }
    int i=2,temp=a[1];
    while(i<=n){
    if(i<n&&a[i]>a[i+1])
    {
    i=i+1;
    }
    if(a[i]<temp)
    {
    a[i/2]=a[i];
    i=i*2;
    }
    else
    {
    break;
    }
    }
    a[i/2]=temp;
    }
    // 4) Main driver code
    int main(){
```

```
// 4) Main driver code
int main(){
int a[10000],n,i,temp;
double startTime,endTime;
printf("\nEnter the value of n : ");
scanf("%d",&n);
printf("enter the elements to be sorted\n");
for(i=1;i<=n;i++){
scanf("%d",&a[i]);
}
startTime = clock();
heap(a,n);
for(i=n;i>=2;i--){
temp=a[1];
a[1]=a[i];
a[i]=temp;
adjust(a,i-1);
}
endTime = clock();
printf("\n After sorting:\n");
for(i=1;i<=n;i++)
printf("%d\t",a[i]);
printf("\nTimetaken is %f\n ",(((double)
(endTime-startTime))/CLOCKS_PER_SEC));
}
```