

HEAP SORT PROGRAM :-

```
#include <stdio.h>
#include <time.h>

void main(){

    clock_t start,end;
    double cpu_time_taken;
    int n,i;

    printf("enter number of elements\n");
    scanf("%d", &n);

    int a[n] ;
    printf("chosen array is:");
    for (int i = 0; i < n; i++)
    {
        a[i]=rand()%100;

        printf("%d\t",a[i]);

    }
    printf("\n");
    start=clock();
    heap_sort(a,n);
    for (int c = 1; c <= 8000; c++) for (int d = 1; d <= 8000; d++) { }
    end = clock();
    cpu_time_taken = (double)(end - start) / CLOCKS_PER_SEC;
    printf("sorted array:");
```

```
for(int i=0;i<n;i++)  
  
printf("%d\t",a[i]);  
printf("\n");  
printf("time spent: %f sec\n",cpu_time_taken);  
getch();  
  
}
```

```
heap_sort(int a[], int n){  
  
int i, temp;  
  
heap_cons(a,n);  
  
for(int i=n-1;i>0;i--)  
  
{  
  
temp=a[0];  
  
a[0]=a[i];  
  
a[i]=temp;  
  
heap_adj(a,i);
```

```
heap_adj(a,i);
```

```
}
```

```
}
```

```
void heap_adj(int a[], int n)
```

```
{
```

```
int i,j,item;
```

```
j=0;
```

```
item=a[j];
```

```
i=2*j+1;
```

```
while(i<=n-1)
```

```
{
```

```
if(i+1<=n-1)
```

```
{
```

```
if(a[i]<a[i+1])
```

```
i++;
```

```
}
```

```
if(item<a[i])
```

```
{
```

```
a[j]=a[i];
```

```
j=i;
```

```
i=2*j+1;
```

```
}
```

```
else
```

```
break;
```

```
}
```

```
a[j]=item;
```

```

}

void heap_cons(int a[], int n)
{
    int i,j,k,item;
    for(k=1;k<n;k++)
    {
        item=a[k];
        i=k;
        j=(i-1)/2;
        while(i>0 && item>a[j])
        {
            a[i]=a[j];
            i=j;
            j=(i-1)/2;
        }
        a[i]=item;
    }
}

```

```

enter number of elements
100
chosen array is:83      86      77      15      93      35      86      92      49      21      62      27      90      59      63
26      40      26      72      36      11      68      67      29      29      82      73      30      21      62      19      23      84      67      35      29      24      15
22      58      69      67      93      56      11      42      29      73      21      19      84      25      37      98      27      24      36      15
70      13      26      91      80      56      73      62      70      96      81      5      25      84      27      36      8
46      29      13      57      24      95      82      45      14      67      34      64      43      50      87      8
76      78      88      84      3      51      54      99      32      60      76      68      39      12      26      86      94
39
sorted array:2  3      5      5      8      11      11      12      13      13      14      15      15      19      21      21
22      23      24      24      25      26      26      26      26      27      27      29      29      29      29      30      32
34      35      35      36      36      37      39      39      40      42      43      45      46      49      50      51      54
56      56      57      58      59      60      62      62      62      63      64      67      67      67      67      68      6
69      70      70      72      73      73      76      76      77      78      80      81      82      82      83      84      8
84      84      86      86      86      87      88      90      91      92      93      93      94      95      96      98      99
time spent: 0.140029 sec

```