MERGE SORT

```
import java.util.Random;
import java.util.Scanner;
public class MergeSort
{
public static void main(String[] args)
{
int a[]= new int[100000];
Scanner in = new Scanner(System.in);
long start, end;

   System.out.print("Enter the number of
elements to be sorted: ");

int n = in.nextInt();

Random rand= new Random();

for(int i=0;i<n;i++)

a[i]=rand.nextInt(2000);

System.out.println("Array elements to be
sorted are:");

for(int i=0; i<n; i++)

System.out.println(a[i]);

start=System.nanoTime();

mergesort(a,0,n-1);

end=System.nanoTime();

System.out.println("\nThe sorted elements are:
");

for(int i=0; i<n; i++)

System.out.println(a[i]);

System.out.println("\nThe time taken to sort is
"+(end-start)+"ns");

}

static void mergesort(int a[], int low, int high)
{
int mid;

if(low < high)

{

mid = (low+high)/2;

mergesort(a, low, mid);

mergesort(a, mid+1, high);

merge(a, low, mid, high);

}

}

static void merge(int a[], int low, int mid, int
high)

{

int i, j, h, k, b[]= new int[100000];

h=low; i=low; j=mid+1;

while((h<=mid) && (j<=high))

{

if(a[h] < a[j])

{

b[i]=a[h];

h=h+1;

}

else

{

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

MERGE SORT

```
j=j+1;
}
i = i+1;
}
if(h > mid)
{
for(k=j; k<=high; k++)
{
b[i]=a[k];
i= i+1;
}
}
else
{
for(k=h;k<=mid;k++)
{
b[i]=a[k];
i= i+1;
}
}
for(k=low; k<= high; k++)
a[k] = b[k];
}
}
```