**PROGRAM 8: RECURSIVE MERGE SORT**

CODE:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void combine(int [], int, int, int);

void split(int [],int, int);

int main()

{

clock\_t start,end;

double time;

int arr[100000], n, i;

printf("Enter the size of the array:\n");

scanf("%d", &n);

printf("Enter the elements in the array:\n");

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

{

arr[i] = rand()%100;

printf("%d ",arr[i]);

}

start = clock();

split(arr, 0, n - 1);

end = clock();

time = ((double)(end - start))/CLOCKS\_PER\_SEC;

printf("\nTime taken : %lf\n",time);

printf("\nThe sorted list in ascending order after applying MergeSort is:\n");

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

{

printf("%d ", arr[i]);

}

return 0;

}

void split(int arr[],int low,int high)

{

int mid;

if(low < high)

{

mid = (low + high) / 2;

split(arr, low, mid);

split(arr, mid + 1, high);

combine(arr, low, mid, high);

}

}

void combine(int arr[],int low,int mid,int high)

{

int i, m, k, l, temp[100000];

l = low;

i = low;

m = mid + 1;

while ((l <= mid) && (m <= high))

{

if (arr[l] <= arr[m])

{

temp[i] = arr[l];

l++;

}

else

{

temp[i] = arr[m];

m++;

}

i++;

}

if (l > mid)

{

for (k = m; k <= high; k++)

{

temp[i] = arr[k];

i++;

}

}

else

{

for (k = l; k <= mid; k++)

{

temp[i] = arr[k];

i++;

}

}

for (k = low; k <= high; k++)

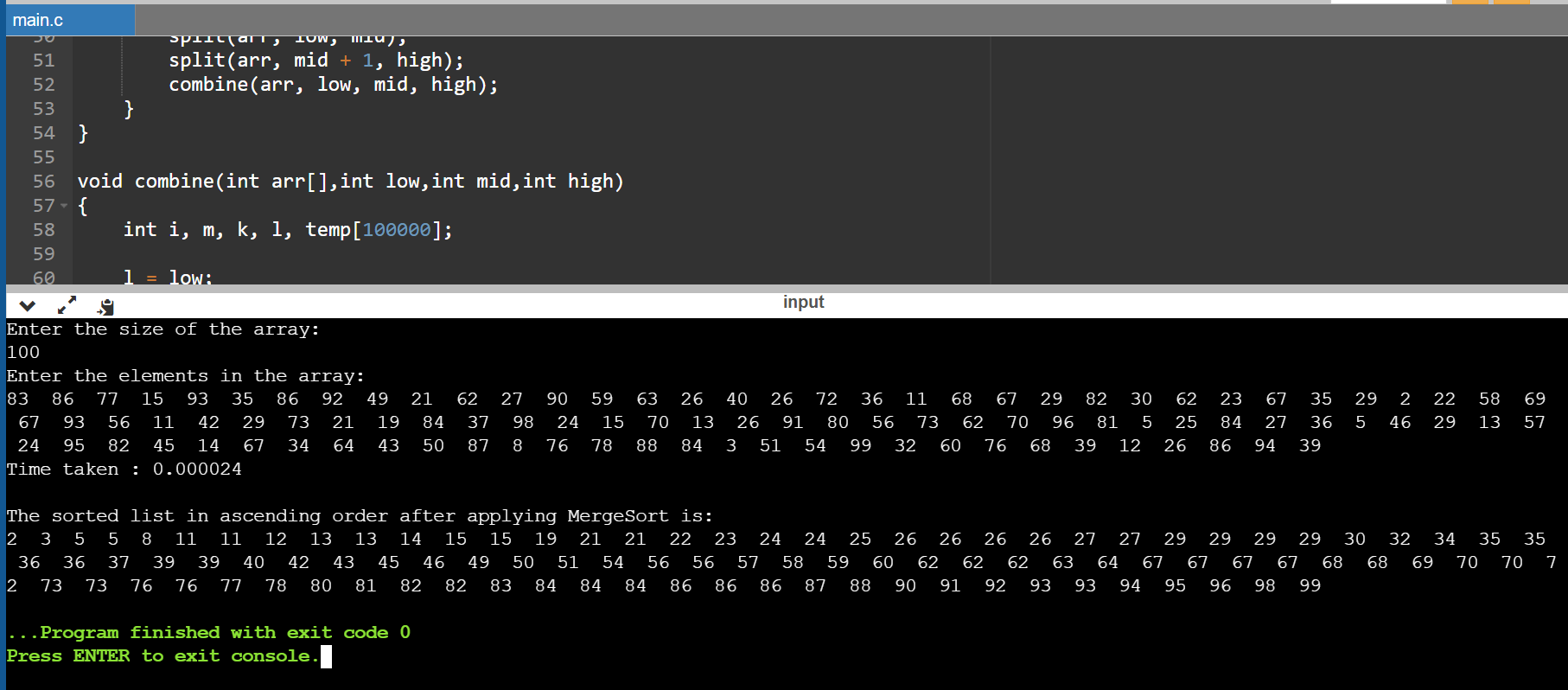
{

arr[k] = temp[k];

}

}

OUTPUT:



|  |  |
| --- | --- |
| N | Time Taken |
| 50 | 0.000017 |
| 100 | 0.000024 |
| 200 | 0.000046 |
| 500 | 0.000111 |
| 1000 | 0.00023 |
| 5000 | 0.000956 |
| 10000 | 0.002056 |
| 15000 | 0.002789 |
| 20000 | 0.004008 |
| 25000 | 0.005103 |