#include <stdio.h>

#include <time.h>

#include <stdlib.h>

void merge(int arr[], int p, int q, int r)

{

int n1 = q - p + 1;

int n2 = r - q;

int L[n1], M[n2];

for (int i = 0; i < n1; i++)

L[i] = arr[p + i];

for (int j = 0; j < n2; j++)

M[j] = arr[q + 1 + j];

int i, j, k;

i = 0;

j = 0;

k = p;

while (i < n1 && j < n2) {

if (L[i] <= M[j]) {

arr[k] = L[i];

i++;

} else {

arr[k] = M[j];

j++;

}

k++;

}

while (i < n1) {

arr[k] = L[i];

i++;

k++;

}

while (j < n2) {

arr[k] = M[j];

j++;

k++;

}

}

void mergeSort(int arr[], int l, int r) {

if (l < r) {

int m = l + (r - l) / 2;

mergeSort(arr, l, m);

mergeSort(arr, m + 1, r);

merge(arr, l, m, r);

}

}

void printArray(int arr[], int size) {

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

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

printf("\n");

}

int main()

{

int n;

clock\_t start, end;

double cpu\_time\_used;

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

scanf("%d",&n);

int arr[n];

printf("The elements of the array:\n");

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

arr[i]= rand()%1000;

printArray(arr, n);

printf("\n");

start = clock();

mergeSort(arr, 0, n-1);

end = clock();

cpu\_time\_used = ((double) (end - start)) / CLOCKS\_PER\_SEC;

printf("Sorted array: \n");

printArray(arr, n);

printf("\n");

printf("TIME FOR FUNCTION EXECUTION is %f", cpu\_time\_used);

return 0;

}

output:

[shilpa@shilpa-linux ADA\_LAB-PROG-main]$ gcc -o merge\_sort merge\_sort.c

[shilpa@shilpa-linux ADA\_LAB-PROG-main]$ ./merge\_sort

Enter the size of the array

100

The elements of the array:

383 886 777 915 793 335 386 492 649 421 362 27 690 59 763 926 540 426 172 736 211 368 567 429 782 530 862 123 67 135 929 802 22 58 69 167 393 456 11 42 229 373 421 919 784 537 198 324 315 370 413 526 91 980 956 873 862 170 996 281 305 925 84 327 336 505 846 729 313 857 124 895 582 545 814 367 434 364 43 750 87 808 276 178 788 584 403 651 754 399 932 60 676 368 739 12 226 586 94 539

Sorted array:

11 12 22 27 42 43 58 59 60 67 69 84 87 91 94 123 124 135 167 170 172 178 198 211 226 229 276 281 305 313 315 324 327 335 336 362 364 367 368 368 370 373 383 386 393 399 403 413 421 421 426 429 434 456 492 505 526 530 537 539 540 545 567 582 584 586 649 651 676 690 729 736 739 750 754 763 777 782 784 788 793 802 808 814 846 857 862 862 873 886 895 915 919 925 926 929 932 956 980 996