LAB 4:MERGE SORT

```
#include <stdio.h>
#include <time.h>
#define max 20

void simplemerge(int a[max], int low, int high, int mid) {
    int j = mid + 1;
    int i = low;
    int k = 0;
    int c[max];
    while (i <= mid && j <= high) {
        if (a[i] < a[j])
            c[k++] = a[i++];
        else
            c[k++] = a[j++];
    }
    while (i <= mid) { c[k++] = a[i++]; }
    while (j <= high) { c[k++] = a[j++]; }
    for (k = 0; k <= high - low; k++) { a[low + k] = c[k]; }
}

void mergesort(int a[max], int low, int high) {
    if (low < high) {
        int mid = (low + high) / 2;
        mergesort(a, low, mid);
        mergesort(a, mid + 1, high);
        simplemerge(a, low, high, mid);
    }
}

int main() {
    int n;
    int a[max];
    int k;
    printf("Enter number of array elements: ");
    scanf("%d", &n);
    printf("Enter array elements: ");
    for (k = 0; k < n; k++) { scanf("%d", &a[k]); }

#include <stdio.h>
#include <time.h>
#define max 20

void simplemerge(int a[max], int low, int high, int mid) {
    int j = mid + 1;
    int i = low;
    int k = 0;
    int c[max];
```

```
    while (i <= mid && j <= high) {
        if (a[i] < a[j])
            c[k++] = a[i++];
        else
            c[k++] = a[j++];
    }
    while (i <= mid) { c[k++] = a[i++]; }
    while (j <= high) { c[k++] = a[j++]; }
    for (k = 0; k <= high - low; k++) { a[low + k] = c[k]; }
}

void mergesort(int a[max], int low, int high) {
    if (low < high) {
        int mid = (low + high) / 2;
        mergesort(a, low, mid);
        mergesort(a, mid + 1, high);
        simplemerge(a, low, high, mid);
    }
}

int main() {
    int n;
    int a[max];
    int k;
    printf("Enter number of array elements: ");
    scanf("%d", &n);
    printf("Enter array elements: ");
    for (k = 0; k < n; k++) { scanf("%d", &a[k]); }

    // Start measuring time
    clock_t start, end;
    double cpu_time_used;

    start = clock();
    mergesort(a, 0, n - 1);
    end = clock();

    cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;

    printf("Sorted array elements: ");
    for (k = 0; k < n; k++) { printf("%d ", a[k]); }
    printf("\n");

    printf("Time taken for sorting: %f seconds\n", cpu_time_used);

    return 0;
}
```

OUTPUT:

```
Enter number of array elements: 7
Enter array elements: 32 673425 855 235 7

45 74
Sorted array elements: 7 32 45 74 235 855 673425
Time taken for sorting: 0.000000 seconds

Process returned 0 (0x0)   execution time : 21.705 s
Press any key to continue.
```