30-5-24

Lab-5

Sort a given set of N integers using Merge sort

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

void split (int[], int, int);
void combine (int[], int, int);
void main()
{
    int a[15000], n, i, j, ch, temp;
    clock_t start, end;
    while(1) {
        printf("\n1: For manual entry of N value and array elements:");
        printf("\n2: To display time taken for sorting number of
                elements N in the range 500 to 14500");
        printf("\n3: To exit");
        printf("\nEnter your choice:");
        scanf("%d", &ch);
        switch(ch) {
            case 1:
                printf("\nEnter the number of elements: ");
                scanf("%d", &n);
                printf("\nEnter array elements: ");
                for(i=0; i<n; i++) {
                    scanf("%d", &a[i]);
                }
                start = clock();
                split(a, 0, n-1);
                end = clock();
                printf("\nSorted array is:");
                for(i=0; i<n; i++) {
                    printf("%d\t", a[i]);
                }
```

```
        printf("\n Time taken to sort %d numbers is %f secs", n,
                ((double)(end-start))/CLOCKS_PER_SEC);
        break;
    case 2:
        n=500;
        while (n<=14500){
            for(i=0; i<n; i++){
                a[i] = n-i;
            }
            start = clock();
            split (a, 0, n-1);
            end = clock();
            printf("\n Time taken to sort %d numbers is %f secs", n,
                ((double)(end-start))/CLOCKS_PER_SEC);
            n+=1000;
        }
        break;
    case 3:
        exit(0);
        break;
    }
    getchar();
}
return 0;
}

void split (int a[], int low, int high){
    int mid;
    if(low<high){
        mid = (low+high)/2;
        split (a, low, mid);
        split (a, mid+1, high);
        combine (a, low, mid, high);
    }
}
```

```
void combine (int a[], int low, int mid, int high) {
    int c[15000], i, j, k;
    i = k = low;
    j = mid + 1;
    while (i <= mid && j <= high) {
        if (a[i] < a[j])
            c[k] = a[i];
            ++k;
            ++i; }
        else
            c[k] = a[j];
            ++k;
            ++j;
    }
    if (i > mid) {
        while (j <= high) {
            c[k] = a[j];
            ++k;
            ++j;
        }
    }
    if (j > high) {
        while (i <= mid) {
            c[k] = a[i];
            ++k;
            ++i;
        }
    }
    for (i = low; i <= high; i++)
    {
        a[i] = c[i];
    }
}
```

Output:

1. For manual entry of N value and array elements
2. To display time taken for sorting number of elements N in the range 500 to 14500.
3. To exit

Enter your choice: 1
Enter the number of elements: 4
Enter array elements: 44 33 22 11
Sorted array is: 11 22 33 44
Time taken to sort 4 numbers is 0.000012 secs

Enter your choice: 2
Time taken to sort 500 numbers is 0.063000 secs
Time taken to sort 1500 numbers is 0.062000 secs
Time taken to sort 2500 numbers is 0.047000 secs

3500	0.047000
4500	0.047000
5500	0.062000
6500	0.016000
7500	0.031000
8500	0.047000
9500	0.016000
10500	0.062000
11500	0.016000
12500	0.031000
13500	0.016000
14500	0.046000