Run Debug Stop Share Save { } Beautify Language C

main.c

```c
#include <stdio.h>
#include <time.h>
#include<stdlib.h>
void merge_sort(int i, int j, int a[], int aux[]) {
    if (j <= i) {
        return;
    }
    int mid = (i + j) / 2;
    merge_sort(i, mid, a, aux);
    merge_sort(mid + 1, j, a, aux);
    int pointer_left = i;
    int pointer_right = mid + 1;
    int k;
    for (k = i; k <= j; k++) {
        if (pointer_left == mid + 1) {
            aux[k] = a[pointer_right];
            pointer_right++;
        } else if (pointer_right == j + 1) {
            aux[k] = a[pointer_left];
            pointer_left++;
        } else if (a[pointer_left] < a[pointer_right]) {
            aux[k] = a[pointer_left];
            pointer_left++;
        } else {
            aux[k] = a[pointer_right];
            pointer_right++;
        }
    }
    for (k = i; k <= j; k++) {
```

Connecting...

Search the web and Windows 23:05 26-05-2021

Run | Debug | Stop | Share | Save | { } Beautify | Language C

main.c

```
        } else {
            aux[k] = a[pointer_right];
            pointer_right++;
        }
    }
    for (k = i; k <= j; k++) {
        a[k] = aux[k];
    }
}
int main() {
  int a[100], aux[100], n, i, d, swap;
  while(1){
  printf("Enter number of elements in the array:\n");
  scanf("%d", &n);
  printf("Random %d integers\n", n);
  for (i = 0; i < n; i++)
    a[i]=1+rand()%1000;
    clock_t begin = clock();
  merge_sort(0, n - 1, a, aux);
  printf("Printing the sorted array:\n");
  for (i = 0; i < n; i++){
     printf("%d\t", a[i]);
}
 clock_t end = clock();
  double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
  printf("\nEXECUTION TIME : %.10fseconds\n\n", time_spent);
  }
  return 0;
}
```

Search the web and Windows

23:06
26-05-2021

input

```
Enter number of elements in the array:
250
Random 250 integers
Printing the sorted array:
12      12      12      12      12      12      13      13      13      23      23      23      23      23      23      28      28      28      2
8       28      28      43      43      43      43      43      43      44      44      44      59      59      59      59      59      59      6
0       60      60      60      60      60      61      61      61      68      68      68      68      68      68      70      70      70      7
0       70      70      85      85      85      85      85      85      88      88      88      92      92      92      92      92      92      9
5       95      95      124     124     124     124     124     124     124     124     124     124     124     124     124     124     124     1
24      124     124     124     124     124     124     124     124     124     124     124     124     124     124     124     124     124     1
24      124     124     124     124     124     124     124     124     124     124     124     124     124     124     124     124     124     1
24      124     124     124     124     124     124     124     124     124     124     124     124     124     124     124     124     124     1
24      124     124     124     124     124     124     124     125     136     136     168     168     171     171     173     173     179     1
99      199     212     212     227     230     230     277     282     282     306     306     314     316     316     325     325     328     3
36      336     337     363     363     365     12      12      13      23      23      28      28      43      43      44      59      59      6
0       60      61      68      68      70      70      85      85      88      92      92      95      369     371     374     384     387     3
94      414     422     422     427     430     457     493     527     12      12      12      12      12      12      13      13      13      2
3       23      23      23      23      23      28      28      28      28      28      28      43      43      43      43
EXECUTION TIME : 0.0000660000seconds

Enter number of elements in the array:
```