PD2 2023

HW2

CSIE@NCKU 2023

Homework Description:

Given a simple csv file (file name will be given by the input argument) in which each line contains the employee information “employee_id”, “age”, “gender”, and “salary”. Could you use concept of CLASSES to answer:

In each line, list all employee_ids which corresponding salaries are the same, and each employee_id is separated by “,”. The first value in each line is the salary. Lines are printed out according to the corresponding salary line by line in the ascending order.
The order of employee_ids in each line should be sorted by the corresponding age in the ascending order. If more than two equal-salary employees have the same age, the order should ‘lady first’.
Please use ‘cout’ to send all result to console.

Deadline:

2023/3/29 23:59 (Wed.).
HW2 should be submitted before the deadline. No excuse to submit your code after the deadline. TA will copy your code at 2023/3/29 0:01. If your code is not in the hw2 Folder, you will get score ‘0’.

Environment:

1. uname -a
   Linux version 5.15.0-67-generic (buildd@lcy02-amd64-116) (gcc (Ubuntu 11.3.0-1ubuntu1~22.04) 11.3.0, GNU ld (GNU Binutils for Ubuntu) 2.38) #74-Ubuntu SMP Wed Feb 22 14:14:39 UTC 2023
2. IP: 140.116.246.230
3. Please remember you should connect to the server with a NCKU IP.

Make sure you use NCKU VPN or connect to the server in our school.

Example:

An example csv file, named test.csv, may be as the form:

```
100123,32,f,50000
100142,28,m,32500
100144,41,f,75000
92301,20,f,50000
101231,41,m,75000
```

Input:

```
test.csv
```

Output:

```
32500,100142
50000,92301,100123
75000,100144,101231
```

Note:

1. It is encouraged that you can “use chatGPT”.
2. The maximum number of employees is smaller than 10,000, meaning that the number of input line is smaller than 10,000.
3. Remember the strict ordering policy. We will examine your correctness by our shell script without any excuse.
4. Please list the result to console. We will use the shell operator “>” to copy all your homework output to a created file named “result”.
5. You need to declare the executable file named “**hw2**” which will be generated by using your makefile with “make all”.
6. The csv file name will be given as the input argument without any exception of file handle error.
7. For the convenience of checking your homework, **output** must not contain any blank characters or other characters, otherwise your score will be deducted.
8. It is okay if you know how to use the STL library.
9. We only count the result correction. The execution efficiency is relatively minor in this homework (unless the execution hangs more than 10 minutes).

10. If you know how to use scp (you may use Windows-based PowerShell), you could try scp in powershell like:

```
scp hw2.cpp ktchuang@140.116.246.230:~/hw2
```

How to Submit:

Please pay attention to the following instructions when submitting homework:

1. Under your account folder, create the folder named "**hw2**".
   (*Please note that you must pay attention to the correct capitalization. If we cannot correctly copy the foler hw2, you will get score '0'.)

2. Put every necessary files under the folder, including :
   i. Your **main program**, such as main.cpp, hw2.cpp, main.h, program.h, etc.
   ii. **makefile** (*Name your executable file as "hw2")
3. Make sure it works normally under the folder and all files are in the correct path.

Examples of files in your folder:

```
netdb@2023pd2:/home/vs6112030/hw2$ pwd
/home/vs6112030/hw2
netdb@2023pd2:/home/vs6112030/hw2$ ls
hw2  main.cpp  makefile
```

How we execute:

```
netdb@2023pd2:/home/vs6112030/hw2$ make all
netdb@2023pd2:/home/vs6112030/hw2$ ./hw2 test.csv > result
```

Supplement about sorting (coached by ChatGPT with GPT-4 engine):

```
#include <iostream>
using namespace std;

// Bubble Sort 函數
void bubbleSort(int arr[], int n) {
    int i, j;
    bool swapped;
    // 外層循環，迭代整個數組
    for (i = 0; i < n - 1; i++) {
        swapped = false;
        // 內層循環，逐個比較相鄰的元素
        for (j = 0; j < n - i - 1; j++) {
            // 如果相鄰元素順序不對，就交換它們
            if (arr[j] > arr[j + 1]) {
                swap(arr[j], arr[j + 1]);
                swapped = true;
            }
        }
        // 如果內部循環沒有進行任何交換，意味著數組已經排序完成，提前退出外部循環
        if (swapped == false) {
            break;
        }
    }
}
int main() {
    int arr[] = { 64, 25, 12, 22, 11 };
    int n = sizeof(arr) / sizeof(arr[0]);
    // 呼叫 Bubble Sort 函數進行排序
    bubbleSort(arr, n);
    cout << "Sorted array: \n";
    // 顯示排序後的數組
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
    return 0;
}
```