CS205 C/ C++ Programming - Lab Assignment4

Name: 郑鑫颖

SID: 11912039

Problem1-3

Part 1- Analysis

This problem include three parts.

Firstly, generate a random score table for the fourteen labs;

Secondly get the student ID who got zero score in more than 1 lab(>=2);

Thirdly, output the data into a csv file.

I implemented 3 functions to realize the functions.

int n;//number of students

int* SID; //Student ID

int* score[14];//Scores

since these arguments are used in all three parts, I defined them globally.

generate ():

I: define **random(x) rand() % (x + 1)** in the beginning to randomly generate the score for each lab. Note that use **(int)time(0)** as seed, ensure that the series differs each time.

II: since the first 4 number donotes the year, so it was generated seperatedly.

SID[i] = (random(20) + 2000) * 10000 + random(8999) + 1000;

(random(20) + 2000)∈[2000,2020]   random(8999) + 1000∈[1000-9999]

III: IDs should be unique. So I use a **O(n^2)** algorithm(tranverse), to regenerated the ID if is repetitive.

absent ():

Count the number of Labs with scores of 0 for each student, and output the student ID if they are greater than or equal to 2.

output ():

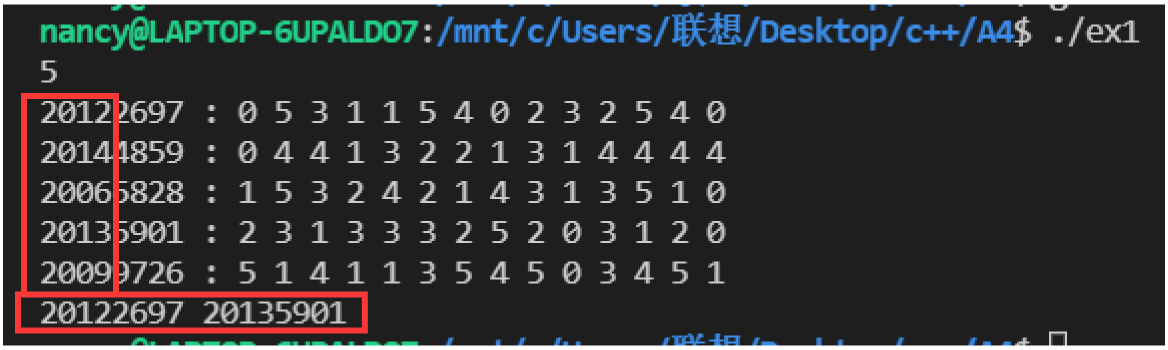
Export the data to lab_records.csv.

Part 2- Code

```
#include <iostream>
#include <ctime>
#include <fstream>
#define random(x) rand() % (x + 1) //0-x

using namespace std;
void absent(int n, int *, int(* score)[14]); //ex2 输出缺席超过两次的学生的学号
void output(int n, int *SID, int(* score)[14]); //ex3 把信息输出
void generate();
int n;
int* SID;
int (*score)[14];

int main()
{
    generate();
    absent(n, SID, score);
    output(n,SID,score);
    delete[] SID;
    delete[] score;
}

void generate()
{
    srand((int)time(0));

    cin >> n;
    bool flag; //判断学号是否重复

    SID=new int[n];
    score=new int[n][14];
    for (int i = 0; i < n; i++)
    {
        flag = true;
        SID[i] = (random(20) + 2000) * 10000 + random(8999) + 1000;

        while (flag)
        {
            flag = false;
            for (int x = 0; x < i; x++)
            {
                if (SID[i] == SID[x])
                {
                    flag = true;
                    break;
                }
            }
            if (flag == true)
            {
                SID[i] = (random(20) + 2000) * 10000 + random(8999) + 1000;
            }
        }
        cout<<SID[i]<<" : ";

        for (int j = 0; j < 13; j++)
        {
            score[i][j] = random(5);
            cout<<score[i][j]<<" ";
        }
        score[i][13]=random(5);
        cout<<score[i][13]<<endl;

    }
}
void absent(int n, int *SID, int(* score)[14])
{
    int t;
    for (int x = 0; x < n; x++)
    {
        t = 0;
        for (int y = 0; y < 14; y++)
        {
            if (score[y][x] == 0)
            {
                t++;
            }
        }
        if (t >= 2)
        {
            cout << SID[x] << " ";
        }
    }
    cout << endl;
}
void output(int n, int *SID, int(* score)[14])
{
    ofstream output;
    output.open("lab_records.csv");
    if (!output.is_open())
    {
        "Can not open the file! ";
    }
    else
    {
        for (int i = 0; i < n; i++)
        {
            output << SID[i] << " : ";
            for (int j = 0; j < 13; j++)
            {
                output << score[i][j] << " , ";
            }
            output << score[i][13] << endl;
        }
    }
    output.close();
}
```

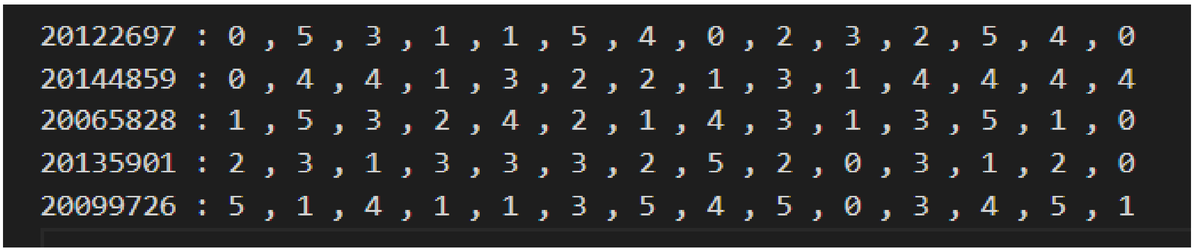
Part 3- Result & Verification

Test case #1: **first four number is in the right range.**



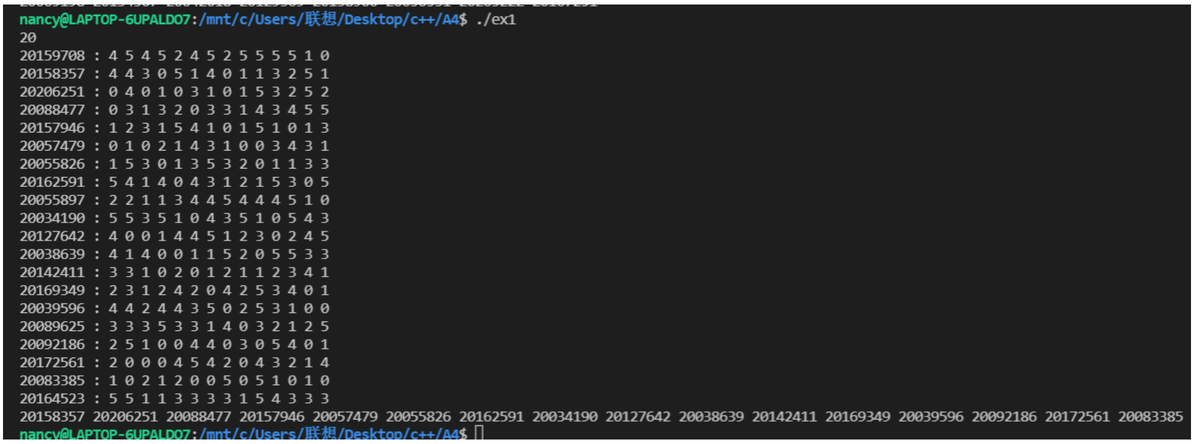
```
nancy@LAPTOP-6UPALDO7:/mnt/c/Users/联想/Desktop/c++/A4$ ./ex1
5
20122697 : 0 5 3 1 1 5 4 0 2 3 2 5 4 0
20144859 : 0 4 4 1 3 2 2 1 3 1 4 4 4 4
20065828 : 1 5 3 2 4 2 1 4 3 1 3 5 1 0
20135901 : 2 3 1 3 3 3 2 5 2 0 3 1 2 0
20099726 : 5 1 4 1 1 3 5 4 5 0 3 4 5 1
20122697 20135901
```

in the csv file:

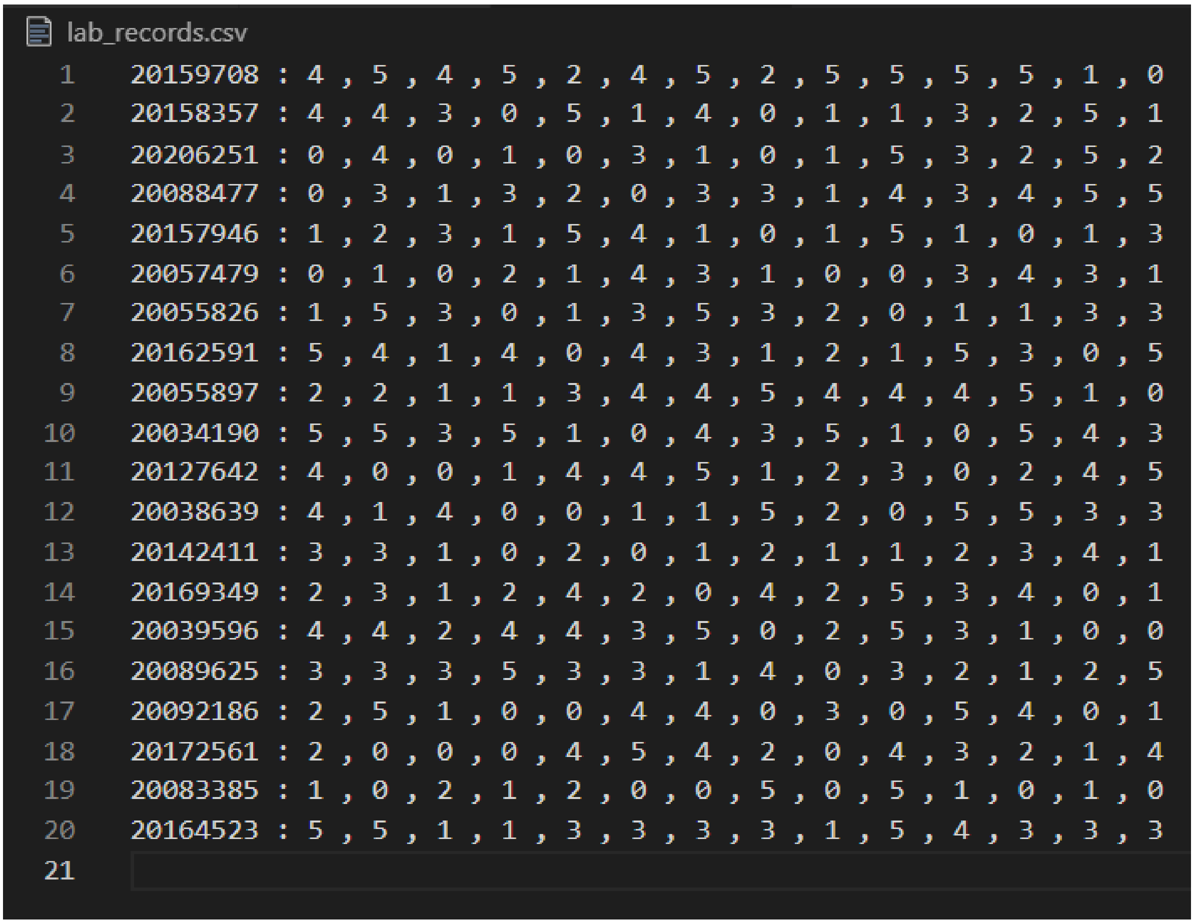


```
20122697 : 0 , 5 , 3 , 1 , 1 , 5 , 4 , 0 , 2 , 3 , 2 , 5 , 4 , 0
20144859 : 0 , 4 , 4 , 1 , 3 , 2 , 2 , 1 , 3 , 1 , 4 , 4 , 4 , 4
20065828 : 1 , 5 , 3 , 2 , 4 , 2 , 1 , 4 , 3 , 1 , 3 , 5 , 1 , 0
20135901 : 2 , 3 , 1 , 3 , 3 , 3 , 2 , 5 , 2 , 0 , 3 , 1 , 2 , 0
20099726 : 5 , 1 , 4 , 1 , 1 , 3 , 5 , 4 , 5 , 0 , 3 , 4 , 5 , 1
```

Test case #2:relatively big example.



```
nancy@LAPTOP-6UPALDO7:/mnt/c/Users/联想/Desktop/c++/A4$ ./ex1
20
20159708 : 4 5 4 5 2 4 5 2 5 5 5 5 1 0
20158357 : 4 4 3 0 5 1 4 0 1 1 3 2 5 1
20206251 : 0 4 0 1 0 3 1 0 1 5 3 2 5 2
20088477 : 0 3 1 3 2 0 3 3 1 4 3 4 5 5
20157946 : 1 2 3 1 5 4 1 0 1 5 1 0 1 3
20057479 : 0 1 0 2 1 4 3 1 0 0 3 4 3 1
20055826 : 1 5 3 0 1 3 5 3 2 0 1 1 3 3
20162591 : 5 4 1 4 0 4 3 1 2 1 5 3 0 5
20055897 : 2 2 1 1 3 4 4 5 4 4 4 5 1 0
20034190 : 5 5 3 5 1 0 4 3 5 1 0 5 4 3
20127642 : 4 0 0 1 4 4 5 1 2 3 0 2 4 5
20038639 : 4 1 4 0 0 1 1 5 2 0 5 5 3 3
20142411 : 3 3 1 0 2 0 1 2 1 1 2 3 4 1
20169349 : 2 3 1 2 4 2 0 4 2 5 3 4 0 1
20039596 : 4 4 2 4 4 3 5 0 2 5 3 1 0 0
20089625 : 3 3 3 5 3 3 1 4 0 3 2 1 2 5
20092186 : 2 5 1 0 0 4 4 0 3 0 5 4 0 1
20172561 : 2 0 0 0 4 5 4 2 0 4 3 2 1 4
20083385 : 1 0 2 1 2 0 0 5 0 5 1 0 1 0
20164523 : 5 5 1 1 3 3 3 3 1 5 4 3 3 3
20158357 20206251 20088477 20157946 20057479 20055826 20162591 20034190 20127642 20038639 20142411 20169349 20039596 20092186 20172561 20083385
nancy@LAPTOP-6UPALDO7:/mnt/c/Users/联想/Desktop/c++/A4$
```



lab_records.csv

```
20159708 : 4 , 5 , 4 , 5 , 2 , 4 , 5 , 2 , 5 , 5 , 5 , 5 , 1 , 0
20158357 : 4 , 4 , 3 , 0 , 5 , 1 , 4 , 0 , 1 , 1 , 3 , 2 , 5 , 1
20206251 : 0 , 4 , 0 , 1 , 0 , 3 , 1 , 0 , 1 , 5 , 3 , 2 , 5 , 2
20088477 : 0 , 3 , 1 , 3 , 2 , 0 , 3 , 3 , 1 , 4 , 3 , 4 , 5 , 5
20157946 : 1 , 2 , 3 , 1 , 5 , 4 , 1 , 0 , 1 , 5 , 1 , 0 , 1 , 3
20057479 : 0 , 1 , 0 , 2 , 1 , 4 , 3 , 1 , 0 , 0 , 3 , 4 , 3 , 1
20055826 : 1 , 5 , 3 , 0 , 1 , 3 , 5 , 3 , 2 , 0 , 1 , 1 , 3 , 3
20162591 : 5 , 4 , 1 , 4 , 0 , 4 , 3 , 1 , 2 , 1 , 5 , 3 , 0 , 5
20055897 : 2 , 2 , 1 , 1 , 3 , 4 , 4 , 5 , 4 , 4 , 4 , 5 , 1 , 0
20034190 : 5 , 5 , 3 , 5 , 1 , 0 , 4 , 3 , 5 , 1 , 0 , 5 , 4 , 3
20127642 : 4 , 0 , 0 , 1 , 4 , 4 , 5 , 1 , 2 , 3 , 0 , 2 , 4 , 5
20038639 : 4 , 1 , 4 , 0 , 0 , 1 , 1 , 5 , 2 , 0 , 5 , 5 , 3 , 3
20142411 : 3 , 3 , 1 , 0 , 2 , 0 , 1 , 2 , 1 , 1 , 2 , 3 , 4 , 1
20169349 : 2 , 3 , 1 , 2 , 4 , 2 , 0 , 4 , 2 , 5 , 3 , 4 , 0 , 1
20039596 : 4 , 4 , 2 , 4 , 4 , 3 , 5 , 0 , 2 , 5 , 3 , 1 , 0 , 0
20089625 : 3 , 3 , 3 , 5 , 3 , 3 , 1 , 4 , 0 , 3 , 2 , 1 , 2 , 5
20092186 : 2 , 5 , 1 , 0 , 0 , 4 , 4 , 0 , 3 , 0 , 5 , 4 , 0 , 1
20172561 : 2 , 0 , 0 , 0 , 4 , 5 , 4 , 2 , 0 , 4 , 3 , 2 , 1 , 4
20083385 : 1 , 0 , 2 , 1 , 2 , 0 , 0 , 5 , 0 , 5 , 1 , 0 , 1 , 0
20164523 : 5 , 5 , 1 , 1 , 3 , 3 , 3 , 3 , 1 , 5 , 4 , 3 , 3 , 3

```

Part 4 - Difficulties & Solutions

The biggest problem apprears when I want to pass the array as an argument for a function. It always has syntax errors. While debug this, I learned a lot.

1) while defining global array, the length of the array must be a constant.BUT inside a function is ok(still confused about this.)

2)for a muti-demension array, only one demension can be unkown

3) **int (*score) [14]:** define a array pointer point to an array which contain 14 pointer data.

int *score[14]: define a pointer to an array which contain 14 elements

Problem 4

Part 1- Analysis

This problem ask us to read in the csv file and get the lab ID whose average score is less than the average score.

First **read in the score data**, then **calculate each lab's average score** and **compare with the course's average score**.

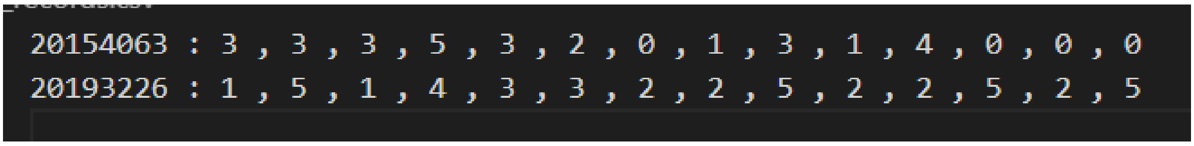
Part 2- Code

```
#include <fstream>
#include <iostream>
#include <cstring>
#include <vector>
using namespace std;
```

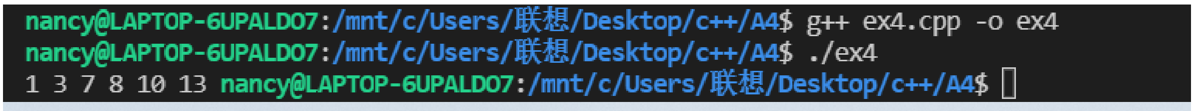
```
int main()
{
    vector<int> SID;
    vector<int> labscore;
    vector<vector<int>> score;
    vector<double>labAver;
    double courseAver=0;
    double labSum=0;

    ifstream input;
    char contents[100];
    char gap[] = ":, ";
    char *token = NULL;
    char *temp = NULL;
    input.open("lab_records.csv");

    if (input.is_open())
    {
        while (!input.eof())
        {
            input.getline(contents, 100);
            if (!contents[0])//error
            {
                break;
            }
            SID.push_back (stoi(strtok_r(contents, gap, &temp)));
            labscore.clear();
            for (int j = 0; j < 14; j++)
            {
                labscore.push_back(stoi(strtok_r(NULL, gap, &temp)));
            }
            score.push_back(labscore);

        }
        for (int j = 0; j < 14; j++)
        {
            for (int i = 0; i < score.size(); i++)
            {
                labSum+=score[i][j];
            }
            labAver.push_back(labSum/score.size());
            courseAver+=labSum;
            labSum=0;
        }
        courseAver/=(14*score.size());

    for (int i = 0; i < 14; i++)
    {
        if (labAver[i]<courseAver)
        {
            cout<<(i+1)<<" ";
        }
    }
    }
    else
    {
        cout << "Cannot open the file!" << endl;
    }
    input.close();
}
```

Part 3- Result & Verification

Test case #1:

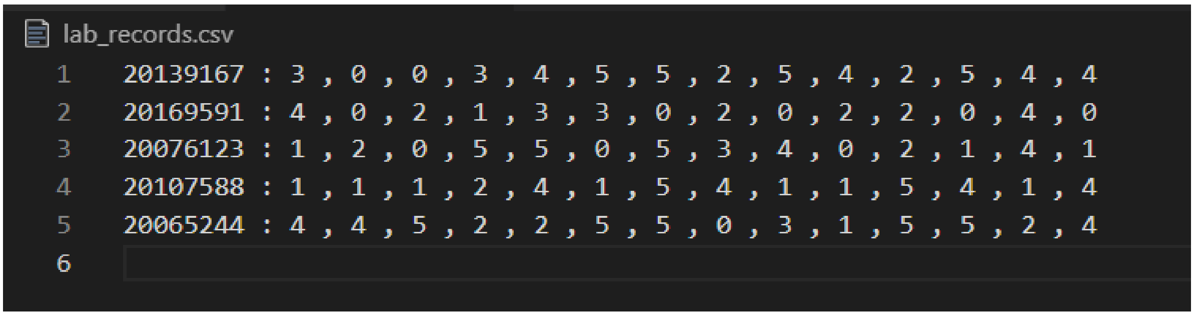


```
20154063 : 3 , 3 , 3 , 5 , 3 , 2 , 0 , 1 , 3 , 1 , 4 , 0 , 0 , 0
20193226 : 1 , 5 , 1 , 4 , 3 , 3 , 2 , 2 , 5 , 2 , 2 , 5 , 2 , 5
```



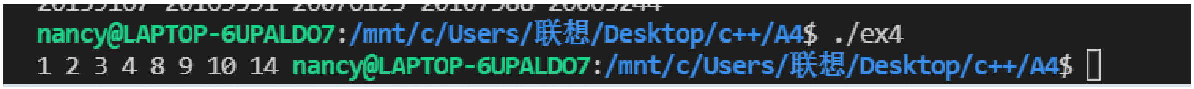
```
nancy@LAPTOP-6UPALDO7:/mnt/c/Users/联想/Desktop/c++/A4$ g++ ex4.cpp -o ex4
nancy@LAPTOP-6UPALDO7:/mnt/c/Users/联想/Desktop/c++/A4$ ./ex4
1 3 7 8 10 13 nancy@LAPTOP-6UPALDO7:/mnt/c/Users/联想/Desktop/c++/A4$
```

Test case #2:



lab_records.csv

```
20139167 : 3 , 0 , 0 , 3 , 4 , 5 , 5 , 2 , 5 , 4 , 2 , 5 , 4 , 4
20169591 : 4 , 0 , 2 , 1 , 3 , 3 , 0 , 2 , 0 , 2 , 2 , 0 , 4 , 0
20076123 : 1 , 2 , 0 , 5 , 5 , 0 , 5 , 3 , 4 , 0 , 2 , 1 , 4 , 1
20107588 : 1 , 1 , 1 , 2 , 4 , 1 , 5 , 4 , 1 , 1 , 5 , 4 , 1 , 4
20065244 : 4 , 4 , 5 , 2 , 2 , 5 , 5 , 0 , 3 , 1 , 5 , 5 , 2 , 4

```



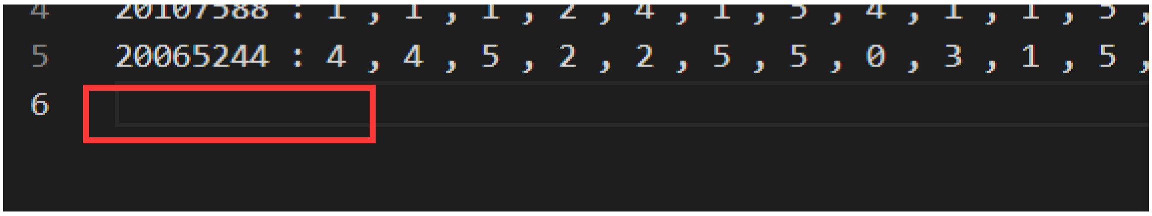
```
20139167 20169591 20076123 20107588 20065244
nancy@LAPTOP-6UPALDO7:/mnt/c/Users/联想/Desktop/c++/A4$ ./ex4
1 2 3 4 8 9 10 14 nancy@LAPTOP-6UPALDO7:/mnt/c/Users/联想/Desktop/c++/A4$
```

Part 4 - Difficulties & Solutions

1) use **stoi** function to convert the string to int.

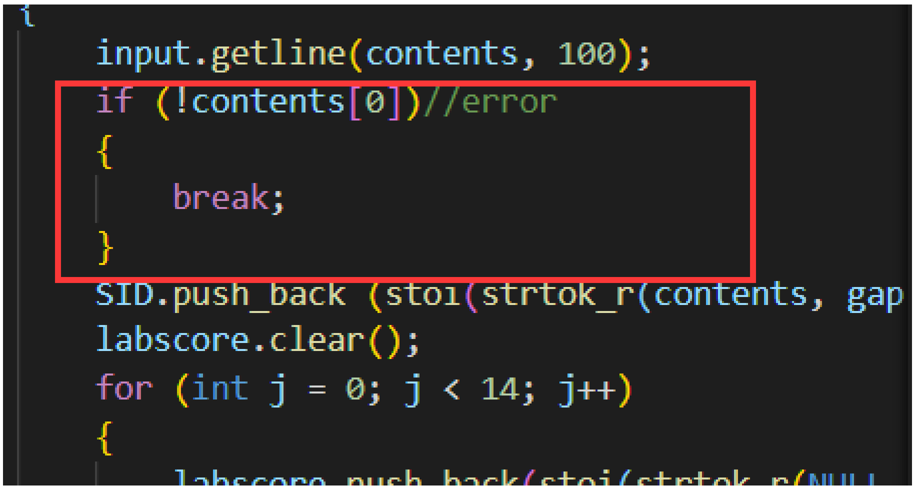
2)Since I don't know the number of students, so instead of array, I use vector to store the information

3)I find that there is always a null string in the end of the csv file, just as the following shows.



```
4    20107588 : 1 , 1 , 1 , 2 , 4 , 1 , 5 , 4 , 1 , 1 , 5 ,
5    20065244 : 4 , 4 , 5 , 2 , 2 , 5 , 5 , 0 , 3 , 1 , 5 ,
6
```

an error will occur in the read in procedure. So I add an error check part;



```
{
    input.getline(contents, 100);
    if (!contents[0])//error
    {
        break;
    }
    SID.push_back (stoi(strtok_r(contents, gap
    labscore.clear();
    for (int j = 0; j < 14; j++)
    {
        labscore.push_back(stoi(strtok_r(NULL
```

if content[0] is null the procedure will break. And the program work well.

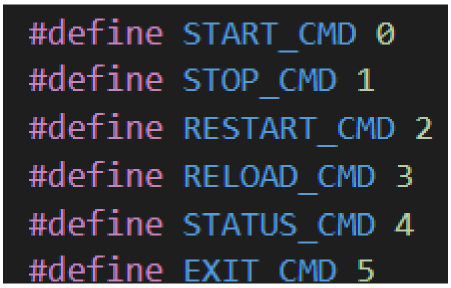
Problem 5

Part 1- Analysis

This problem ask us to recognize the command and display relative information.

To solve this problem, I define in the macro the index of each command, if we change the order in the array, I only need to change the number in the macro.

and in the program, I using switch to match the index with corrosponding operation.



```
#define START_CMD 0
#define STOP_CMD 1
#define RESTART_CMD 2
#define RELOAD_CMD 3
#define STATUS_CMD 4
#define EXIT_CMD 5
```

The program stops only when user enter in "exit" command;

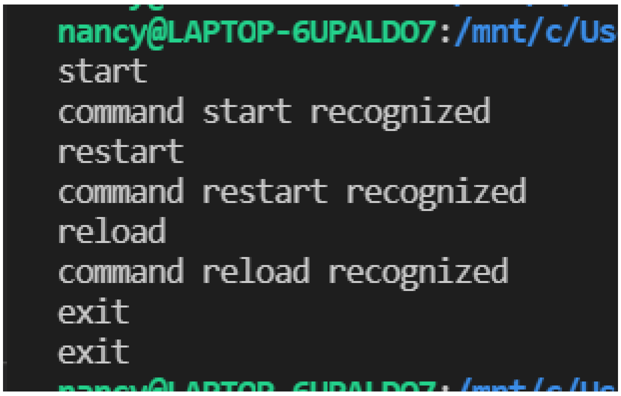
Part 2- Code

My develop environment is:

```
#include<iostream>
using namespace std;
#define START_CMD 0
#define STOP_CMD 1
#define RESTART_CMD 2
#define RELOAD_CMD 3
#define STATUS_CMD 4
#define EXIT_CMD 5

int recognize_CMD();
int main()
{
    while (true)
    {
        int a=recognize_CMD();
        if(a==5) return 0;
        else continue;
    }
}
int recognize_CMD()
{
    const char *commands[]={"start","stop","restart","reload","status","exit"};
    string command;
    int index=-1;
    cin>>command;
    for (int  i = 0; i <6; i++)
    {
        if (command==commands[i])
        {
            index=i;
            break;
        }
    }

    switch (index)
    {
    case START_CMD:
        cout<<"command start recognized"<<endl;
        return START_CMD;
    case STOP_CMD:
        cout<<"command stop recognized"<<endl;
        return STOP_CMD;
    case RESTART_CMD:
        cout<<"command restart recognized"<<endl;
        return RESTART_CMD;
    case RELOAD_CMD:
        cout<<"command reload recognized"<<endl;
        return RELOAD_CMD;
    case STATUS_CMD:
        cout<<"command status recognized"<<endl;
        return STATUS_CMD;
    case EXIT_CMD:
        cout<<"exit"<<endl;
        return EXIT_CMD;

    default:
        cout<<"Invalid command"<<endl;
        return -1;
    }

}
```

Part 3- Result & Verification

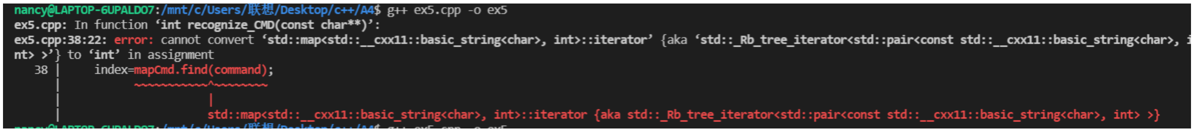
Test case:



```
nancy@LAPTOP-6UPALDO7:/mnt/c/Us
start
command start recognized
restart
command restart recognized
reload
command reload recognized
exit
exit
```

Part 4 - Difficulties & Solutions

I try to use hashmap to match the sting with the index, but meet some errors while using the string as the search key.



```
nancy@LAPTOP-6UPALDO7:/mnt/c/Users/联想/Desktop/c++/A4$ g++ ex5.cpp -o ex5
ex5.cpp: In function ‘int recognize_CMD(const char**)’:
ex5.cpp:38:22: error: cannot convert ‘std::map<std::__cxx11::basic_string<char>, int>::iterator’ {aka ‘std::_Rb_tree_iterator<std::pair<const std::__cxx11::basic_string<char>, i
nt> >’} to ‘int’ in assignment
   38 |     index=mapCmd.find(command);
      |           ~~~~~~~~~~~^~~~~~~~~
      |                      |
      |                      std::map<std::__cxx11::basic_string<char>, int>::iterator {aka std::_Rb_tree_iterator<std::pair<const std::__cxx11::basic_string<char>, int> >}
```