

Basic Description 代码解释

日期:

- 2024年1月3日

环境:

- 操作系统: Windows 11
- 集成开发环境: Microsoft Visual Studio 2022

源代码:

```
#include<stdio.h>
#include<math.h>

// Calculate the BMI value for each person.
static void calc_bmi(float(*infos)[2], float* bmis, char n_people)
{
    /*
    * infos    -> denotes the pointer pointing to the 1D array infos[0];
    * bmis     -> denotes the pointer pointing to the first address of array
    bmis;
    * n_people -> denotes the number of people.
    */
    for (int i = 0; i < (int)n_people; i++) {
        bmis[i] = infos[i][1] / pow((infos[i][0]/100), 2);
    }
}

// Calculate the value of overweight percentage for each person.
static void calc_overweight(float(*infos)[2], char* gender, float* overweight,
char n_people)
{
    /*
    * infos      -> denotes the pointer pointing to the 1D array infos[0];
    * gender     -> denotes the pointer pointing to the gender array;
    * overweight -> denotes the pointer pointing to the overweight array;
    * n_people   -> denotes the number of people.
    */
    float sw[8];
    for (int i = 0; i < (int)n_people; i++) {
        if (gender[i] == 'M') {
            sw[i] = (infos[i][0] - 80) * 0.7;
        }
        else {
            sw[i] = (infos[i][0] - 70) * 0.6;
        }
        overweight[i] = (infos[i][1] - sw[i]) / sw[i]*100;
    }
}
```

```

// Based on the BMI values, sort the bmis and sorted_index arrays through methods
like bubble/selection sorting.
static void ascending_sorting(float* bmis, char* sorted_index, char n_people)
{
    /*
    * bmis      -> denotes the pointer pointing to the first address of array
    bmis;
    * sorted_index -> denotes the pointer pointing to the sorted_index array;
    * n_people   -> denotes the number of people.
    */
    for (int i = 0; i < (int)n_people - 1; i++) {
        for (int j = 0; j < (int)n_people - i - 1; j++) {
            if (bmis[sorted_index[j]] > bmis[sorted_index[j + 1]]) {
                int temp = sorted_index[j];
                sorted_index[j] = sorted_index[j + 1];
                sorted_index[j + 1] = temp;
            }
        }
    }
}

// Display the result in a terminal window
static void display(const char** names, char* gender, float* bmis, float*
overweight, char* sorted_index, char n_people)
{
    /*
    * names      -> denotes the pointer pointing to the names array;
    * gender     -> denotes the pointer pointing to the gender array;
    * bmis       -> denotes the pointer pointing to the bmis array;
    * overweight -> denotes the pointer pointing to the overweight array;
    * sorted_index -> denotes the pointer pointing to the sorted_index array;
    * n_people   -> denotes the number of people.
    */

    printf("\nThe health condition of these people\n\n");
    printf("%8s", "Name");
    printf("%10s", "Gender");
    printf("%10s", "BMI");
    printf("%28s", "health condition");
    printf("%22s", "Overweight\n\n");
    for (int i=0;i<(int)n_people;i++)
    {
        printf("%7s", names[sorted_index[i]]);
        printf("%8c", gender[sorted_index[i]]);
        printf("%16f", bmis[sorted_index[i]]);
        if (bmis[sorted_index[i]]>=35)
            printf("%23s", "Severe obesity");
        else if (bmis[sorted_index[i]]>=30)
            printf("%23s", "Moderate obesity");
        else if (bmis[sorted_index[i]]>=27)
            printf("%23s", "Mild obesity");
        else if (bmis[sorted_index[i]]>=24)
            printf("%23s", "obesity");
        else if (bmis[sorted_index[i]]>=18.5)
            printf("%23s", "Normal range");
        else

```

```

        printf("%23s", "Underweight");
        printf("%20f", overweight[sorted_index[i]]);
        printf(" %%\n\n");
    }
    printf("\n");
}

int main()
{
    const char* names[] = { "Song", "Zhou", "Chen", "Wang", "Zhao", "Yao", "Shen", "Liu" };
    char gender[] = { 'M', 'F', 'M', 'M', 'F', 'M', 'F', 'F' };
    float infos[8][2] = { {177.3, 66.1}, {162.8, 52.9}, {180.6, 103.7}, {172.3, 71.4},
        {183.0, 91.6}, {158.4, 57.2},
        {166.1, 79.0}, {178.4, 85.3} };
    char n_people = 8; //Number of people

    //Calculate the BMI values for each person
    float bmis[8];
    calc_bmi(infos, bmis, n_people);

    //Sort the people according to the BMI values
    char sorted_index[8] = { 0, 1, 2, 3, 4, 5, 6, 7 };
    ascending_sorting(bmis, sorted_index, n_people);

    // Calculate the overweight percentage values
    float overweight[8];
    calc_overweight(infos, gender, overweight, n_people);

    // Display the result in a terminal window
    display(names, gender, bmis, overweight, sorted_index, n_people);

    return 0;
}

```

解释：

变量和数组

- `names[]`：一个字符串指针数组，每个指针指向一个人的名字。
- `gender[]`：一个字符数组，表示每个人的性别（'M'表示男性，'F'表示女性）。
- `infos[8][2]`：一个二维浮点数组，存储每个人的身高和体重。
- `n_people`：一个字符变量，表示人数。
- `bmis[8]`：一个浮点数组，用于存储计算出的每个人的BMI值。
- `sorted_index[8]`：一个字符数组，用于在排序过程中存储BMI排序后的索引。
- `overweight[8]`：一个浮点数组，用于存储每个人的超重百分比。

函数

1. `calc_bmi`:

- 功能: 为每个人计算BMI值。
- 参数: 指向 `infos[0]` 的指针 (身高和体重数据)、指向BMI数组的指针、人数 `n_people`。
- 算法: 使用标准BMI公式 (体重除以身高的平方 (米为单位)) 计算。

2. `calc_overweight`:

- 功能: 为每个人计算超重百分比。
- 参数: 指向 `infos[0]` 的指针、性别数组、超重数组、人数 `n_people`。
- 算法: 根据性别确定标准体重 (男性为 $(\text{身高cm} - 80) * 0.7$, 女性为 $(\text{身高cm} - 70) * 0.6$), 然后计算超重百分比。

3. `ascending_sorting`:

- 功能: 基于BMI值对 `bmis` 和 `sorted_index` 数组进行排序。
- 参数: BMI数组、排序后的索引数组、人数 `n_people`。
- 算法: 使用冒泡排序算法对BMI值进行升序排序。

4. `display`:

- 功能: 在终端窗口显示结果。
- 参数: 姓名数组、性别数组、BMI数组、超重数组、排序后的索引数组、人数 `n_people`。
- 描述: 根据排序后的索引, 按顺序显示每个人的姓名、性别、BMI、健康状况和超重百分比。

main 函数

- 初始化变量和数组, 包括人名、性别、身高和体重信息。
- 调用 `calc_bmi` 计算每个人的BMI。
- 调用 `ascending_sorting` 对BMI值进行排序。
- 调用 `calc_overweight` 计算每个人的超重百分比。
- 调用 `display` 在终端显示最终结果。