**《面向对象程序设计语言》作业（3.3）**

地信班109092023XXX 许愿

1. 利润提成计算。

#include <iostream>

using namespace std;

int main() {

    int i, temp\_i;

    cout << "请输入当月利润i(元): ";

    cin >> i;

    temp\_i = i; // 本变量留给写法4使用

    // 写法1 直接加和

    double total\_salary;

    if (i <= 100000) {

        total\_salary = i \* 0.1;

    } else if (i <= 200000) {

        total\_salary = 10000 + (i - 100000) \* 0.075;

    } else if (i <= 400000) {

        total\_salary = 10000 + 7500 + (i - 200000) \* 0.05;

    } else if (i <= 600000) {

        total\_salary = 10000 + 7500 + 10000 + (i - 400000) \* 0.03;

    } else if (i <= 1000000) {

        total\_salary = 10000 + 7500 + 10000 + 6000 + (i - 600000) \* 0.015;

    } else {

        total\_salary = 10000 + 7500 + 10000 + 6000 + 6000 + (i - 1000000) \* 0.01;

    }

    cout << "应发奖金总数为: " << total\_salary << endl;

    // 写法2 使用switch直接加和

    double total\_salary\_switch;

    switch (i / 100000) {

        case 0: total\_salary\_switch = i \* 0.1; break;

        case 1: total\_salary\_switch = 10000 + (i - 100000) \* 0.075; break;

        case 2:

        case 3: total\_salary\_switch = 10000 + 7500 + (i - 200000) \* 0.05; break;

        case 4:

        case 5: total\_salary\_switch = 10000 + 7500 + 10000 + (i - 400000) \* 0.03; break;

        case 6:

        case 7:

        case 8:

        case 9: total\_salary\_switch = 10000 + 7500 + 10000 + 6000 + (i - 600000) \* 0.015; break;

        default: total\_salary\_switch = 10000 + 7500 + 10000 + 6000 + 6000 + (i - 1000000) \* 0.01; break;

    }

    cout << "应发奖金总数为: " << total\_salary\_switch << endl;

    // 写法3 将每个部分的奖金单独计算

    double total\_salary\_3;

    if (i > 1000000) {

        total\_salary\_3 += (i - 1000000) \* 0.01;

        i = 1000000;

    }

    if (i > 600000) {

        total\_salary\_3 += (i - 600000) \* 0.015;

        i = 600000;

    }

    if (i > 400000) {

        total\_salary\_3 += (i - 400000) \* 0.03;

        i = 400000;

    }

    if (i > 200000) {

        total\_salary\_3 += (i - 200000) \* 0.05;

        i = 200000;

    }

    if (i > 100000) {

        total\_salary\_3 += (i - 100000) \* 0.075;

        i = 100000;

    }

    total\_salary\_3 += i \* 0.1;

    cout << "应发奖金总数为: " << total\_salary\_3 << endl;

    // 写法4 将每个部分的奖金单独计算的优化写法，使用二维数组

    double total\_salary\_4;

    double bonus[6][2] = {

        {1000000, 0.01},

        {600000, 0.015},

        {400000, 0.03},

        {200000, 0.05},

        {100000, 0.075},

        {0, 0.1}

    };

    for(int it = 0; it < 6; it++) {

        if (temp\_i > bonus[it][0]) {

            total\_salary\_4 += (temp\_i - bonus[it][0]) \* bonus[it][1];

            temp\_i = bonus[it][0];

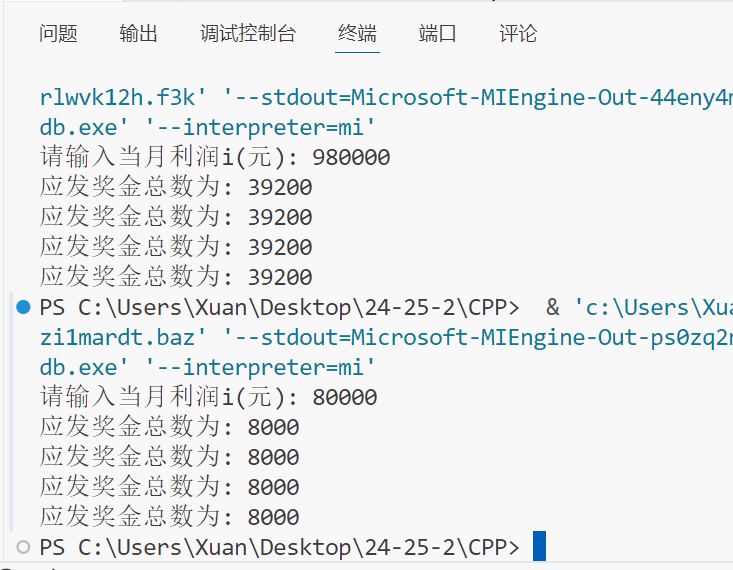
        }

    }

    cout << "应发奖金总数为: " << total\_salary\_4 << endl;

    return 0;

}



1. 输入4个整数，用从小到大的顺序输出。

#include <iostream>

#include <list>

using namespace std;

int main(){

    // 写法1

    list<int> l; // 好像用列表更方便

    for (int i=1; i<=4; i++) { // 初始化i为1，每次循环后自增，即1、2、3、4共获取4个数字

        cout << "请输入第" << i << "个数: ";

        int s; // s作为临时变量

        cin >> s;

        l.push\_back(s); // 将得到的数字加入到列表中

    }

    l.sort();

    cout << "按照从小到大的顺序输出的值为: ";

    for (int i2: l) {

        cout << i2 << " ";

    };

    cout << endl;

    // 写法2 不断交换变量的swap写法

    int a,b,c,d;

    cout << "请输入第1个数: ";

    cin >> a;

    cout << "请输入第2个数: ";

    cin >> b;

    cout << "请输入第3个数: ";

    cin >> c;

    cout << "请输入第4个数: ";

    cin >> d;

    // 获得a,b,c,d

    while (!(a<b && b<c && c<d)) {

        if (a>b) swap(a,b);

        if (b>c) swap(b,c);

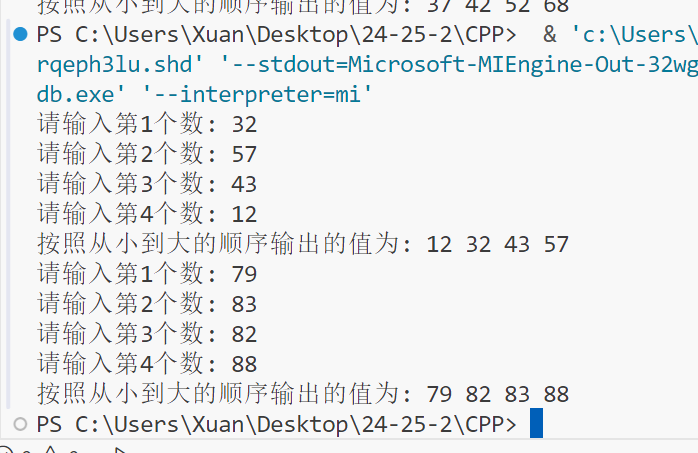
        if (c>d) swap(c,d);

    }

    cout << "按照从小到大的顺序输出的值为: " << a << " " << b << " " << c << " " << d << endl;

    return 0;

}



附加. 计算1/1 + 1/2 + 1/3 + ... + 1/10。

#include <iostream>

using namespace std;

int main() {

    int i = 1;

    float total = 0.0;

    while (i <= 10) { // 从1一直循环到10

        total += 1.0/i; // 使用float防止出现整数问题

        i++;

    }

    cout << "加和得到的值为: " << total << endl;

    return 0;

}

