

实验四、循环语句实验

计算机学院 熊明 20305055

一、实验目的

1. 掌握新的选择语句switch和循环结构
2. 掌握do...while、for语句实现循环结构
3. 理解for、while、do...while三者的区别
4. 掌握转向语句break、continue
5. 掌握循环语句实现延时的原理和方法

二、实验原理

无

三、实验内容

1. 实现1到n的阶乘和

```
1  #include<windows.h>
2  #include<iostream>
3  using namespace std;
4  long long fun(int n) {
5      if (n == 1) return 1; //递归初始值为1
6      else return n * fun(n - 1); //每次返回n乘下一项的返回值
7  }
8  int main() {
9      int n;
10     long long sum = 0; //用long long存防止n值取的过大导致超过int类型的精度
11     while (1) {
12         cin >> n;
13         //if (n == EOF) break; 尝试完成输入ctrl z结束循环, 但失败了
14         for (int i = 1; i <= n; i++) {
15             sum += fun(i); //从1开始累加
16         }
17         cout << sum << endl; //输出累加的值
18     }
19
20 }
```

代码出错, 每次循环未能将sum重置为0,

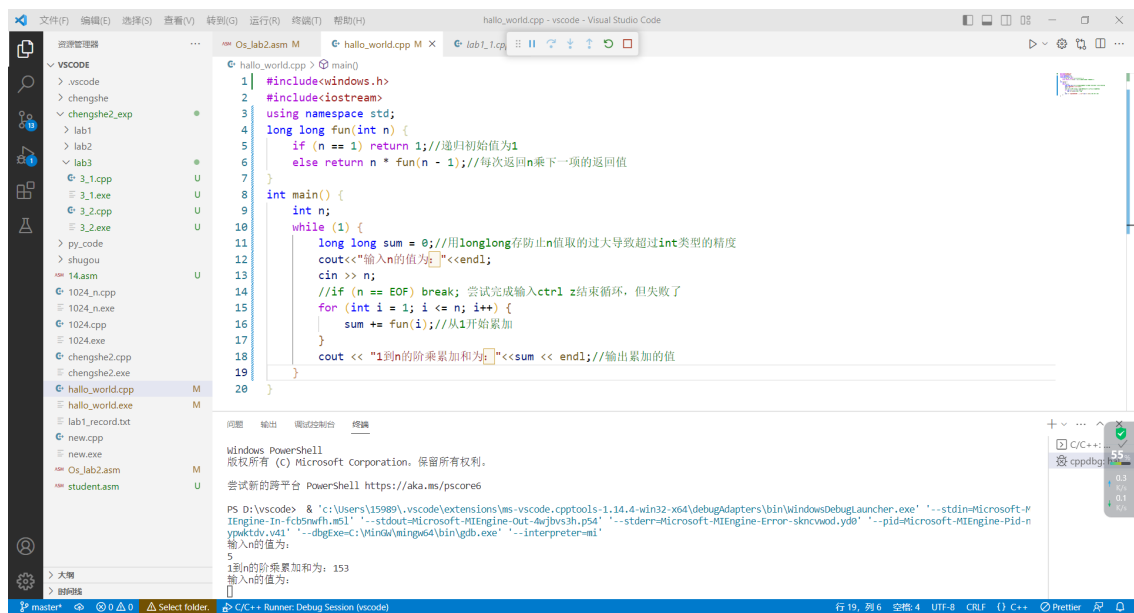
```
1  #include<windows.h>
2  #include<iostream>
3  using namespace std;
4  long long fun(int n) {
5      if (n == 1) return 1; //递归初始值为1
6      else return n * fun(n - 1); //每次返回n乘下一项的返回值
```

```

7   }
8   int main() {
9       int n;
10      while (1) {
11          long long sum = 0; //用longlong存防止n值取的过大导致超过int类型的精度
12          cout<<"输入n的值为: "<<endl;
13          cin >> n;
14          //if (n == EOF) break; 尝试完成输入ctrl z结束循环, 但失败了
15          for (int i = 1; i <= n; i++) {
16              sum += fun(i); //从1开始累加
17          }
18          cout << "1到n的阶乘累加和为: "<<sum << endl; //输出累加的值
19      }
20  }

```

实验结果：



2. 别碰方块游戏

按照ppt内容，分为以下板块：

- 问题分析
- 方块的绘制和移动
- 按空格键控制小球起跳
- 小球和方块的碰撞判断
- 随机方块的速度和高度
- 得分的计算和显示
- 避免空中起跳

代码如下：

```

1   #include<iostream>
2   #include<ctime>
3   #include<conio.h>
4   #include<graphics.h>
5   using namespace std;
6   int main() {
7       //游戏画面的建立
8       float width, height, gravity; //游戏画面的宽、高、重力加速度
9       float ball_x, ball_y, ball_vy, radius; //小球圆心坐标、y方向速度、半径

```

```

10     int score=0,flag=1;//设置分数，flag标志位
11     int isBonF = 1;//标志位，判断小球是否在地面
12     width = 600;//游戏画面宽度
13     height = 400;//游戏画面高度
14     gravity = 0.6;//重力加速度
15     initgraph(width, height);//新建画布
16     //小球的初始化
17     radius = 20;//小球半径
18     ball_x = width / 4;//小球x位置
19     ball_y = height - radius;//小球y位置
20     ball_vy = 0;//小球初始y速度为0
21     //方块的绘制和移动
22     float rec_l_x, rec_t_y, rec_width, rec_height, rec_vx;//方块的参数
23
24     rec_vx = -3;//初始速度为-3;
25     rec_height = 100;//方块高度
26     rec_width = 20;//方块宽度
27     rec_t_y = height - rec_height;//方块顶部y坐标
28     rec_l_x = width * 3 / 4;//方块左边x坐标
29     srand(time(NULL));//通过系统时间来随机设置随机种子
30     while (1) {
31         if (_kbhit()) { //当按键时
32             char input = _getch();//获得输入字符
33             if (input == ' ' && isBonF == 1) { //当按空格并且小球在地面时
34                 ball_vy = -17;//将小球速度置为-17
35                 isBonF = 0;//地面标志位置为0
36             }
37             //if (input == ' ') { //当按下空格
38                 //ball_vy = -16;//可更改，下落速度
39             //}
40         }
41         ball_vy += gravity;//根据重力加速度更新小球y方向速度
42         ball_y += ball_vy;//根据小球y方向速度更新其y坐标
43         if (ball_y >= height - radius) { //如果落在地面上
44             ball_vy = 0;//y速度为0
45             ball_y = height - radius;//规范其y坐标，避免落到地上
46             isBonF = 1;
47         }
48         if (ball_y <= radius) ball_y = radius;//规范其y坐标，避免超过画面顶部
49         //移动方块
50         if (rec_l_x <= -20) { //方块右边线走到最左边时
51             rec_l_x = width;//将方块置于最右边
52             rec_height = rand() % int(height / 4) + height / 4;//随机高度
53             rec_vx = rand() / float(RAND_MAX) * 4 - 7;//随机速度
54             if (flag == 1) { //如果flag位为1，则加分，如果flag位为0，说明该方块被碰撞，不加分
55                 score++;//加分
56             }
57             else flag = 1;//flag标志位置1
58         }
59         rec_l_x += rec_vx;
60         if ((rec_l_x <= ball_x + radius) && (rec_l_x + rec_width >= ball_x - radius) && (height - rec_height <= ball_y + radius)) {
61             sleep(100);//慢动作效果

```

```

62         score = 0; //清零
63         flag = 0; //标志位置0
64     }
65     cleardevice();
66     TCHAR s[20]; //定义字符串数组
67     swprintf_s(s, _T("%d"), score); //不兼容_sprintf函数, 改用
    swprintf_s, 用来将score转换为字符串
68     settextstyle(40, 0, _T("宋体")); //设置文字大小、字体
69     outtextxy(50, 30, s); //输出得分文字
70     fillcircle(ball_x, ball_y, radius); //画圆
71     fillrectangle(rec_l_x, height - rec_height, rec_l_x + rec_width,
    height); //画方块
72     Sleep(10); //延迟10ms
73 }
74 closegraph(); //关闭画布
75
76 }

```

改进的点:

1. 添加flag符号位, 碰到过的方块不会计入分数
2. 添加了随机数种子, 获得真正随机数
3. 修改了_sprintf()函数

实验结果放在压缩包内。

3. 简单计算器的实现

- 设计一个简单计算器, 实现两个数加减乘除取余运算。
- 给出菜单选择

基本思路为:

1. 用循环do while进行重复计算
2. 循环体内部先显示菜单, 提示用户进行各种输入选择
3. 循环体内部先用if语句判断用户输入是否合法, 合法就进行选择的运算
4. 循环体内部菜单选择用switch选择语句 (多种选择)
5. Switch选择语句内部加减乘除每一项计算功能的实现

代码如下:

```

1  #include<iostream>
2  using namespace std;
3  int main(){
4      double x,y;
5      int chose,isFlag = 1;
6      do{
7          cout<<"*****计算器*****"<<endl;
8          cout<<"*****1.加法*****"<<endl;
9          cout<<"*****2.减法*****"<<endl;
10         cout<<"*****3.乘法*****"<<endl;
11         cout<<"*****4.除法*****"<<endl;
12         cout<<"*****5.退出*****"<<endl;
13         cout<<"*****请选择1-5: *****"<<endl;
14         cin>>chose;
15         if(chose>5 || chose<1){
16             cout<<"非法输入, 请重试"<<endl;
17         }else{

```

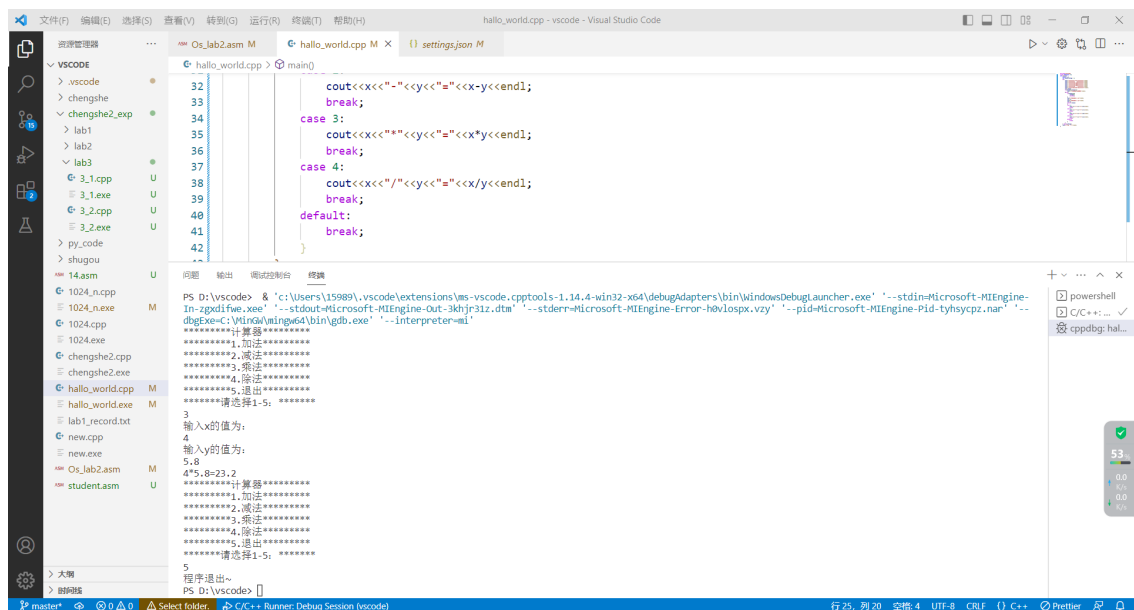
```

18         if(chose==5){
19             isFlag=0;
20             continue;
21         }
22         cout<<"输入x的值为: "<<endl;
23         cin>>x;
24         cout<<"输入y的值为: "<<endl;
25         cin>>y;
26         switch (chose)
27         {
28             case 1:
29                 cout<<x<<"+"<<y<<"="<<x+y<<endl;
30                 break;
31             case 2:
32                 cout<<x<<"-"<<y<<"="<<x-y<<endl;
33                 break;
34             case 3:
35                 cout<<x<<"*"<<y<<"="<<x*y<<endl;
36                 break;
37             case 4:
38                 cout<<x<<"/"<<y<<"="<<x/y<<endl;
39                 break;
40             default:
41                 break;
42         }
43     }
44 } while(isFlag);
45 cout<<"程序退出~"<<endl;
46 }

```

优化的地方：

将输入x和y的值放在switch之前，这样能减少重复代码，但要在输入5时提前加一个判断，若判断输入为5时直接将标志位置为0并且跳过该次循环。



四、实验心得体会

在完成一项工程时，可以先将该工程拆分成不同的小步骤分别完成。

