|  |  |
| --- | --- |
| **编号:** | **1-1** |

****

信息科学与工程学院实验报告

《面向对象程序设计》

**Object-Oriented Programming**

|  |  |
| --- | --- |
| 姓名： | 姬彬荃 |
| 学号： | 201911020125 |
| 班级： | 计联培1901 |
| 时间： | 2020年10月13日 |

**《面向对象程序设计》实验报告**

**基本要求：**请围绕实验目的、实验内容、实验过程、实验结果（附图）、实验总结（重点阐述）五个部分进行撰写。若报告中若涉及源代码内容，请在附录部分提供完整源码及GitHub源码托管地址。报告撰写完毕后请提交PDF格式版本到云班课。

1. **实验目的**

* 理解c++对C的各项改进和扩展基本原理
* 熟练运用C++特色函数解决实际问题
* 理解并掌握C++指针和引用的本质机理
* 熟练掌握C++动态内存申请和释放方法
* 掌握visual studio 代码调试方法

1. **实验内容**

【编程设计题】 给定m根木棍，每根木棍的长度记为L\_i,(3<= i <=m), 下面欲从这m根木棍中选择3根木棍组成周长尽可能最长的三角形、面积尽可能最大的三角形，分别输出最大的周长和面积。如果怎么选都无法构成三角形，请直接输出0。

要求：

1. 算法具有良好的可读性、稳健性和通用性（适合整数长度，浮点数长度）。
2. 给出算法的复杂度分析，算法复杂度尽可能越低越好。
3. 算法设计时采用指针，引用，重载函数，及动态内存申请等C++核心特性。

输入：

m = 5

L = 2 3 4 5 10

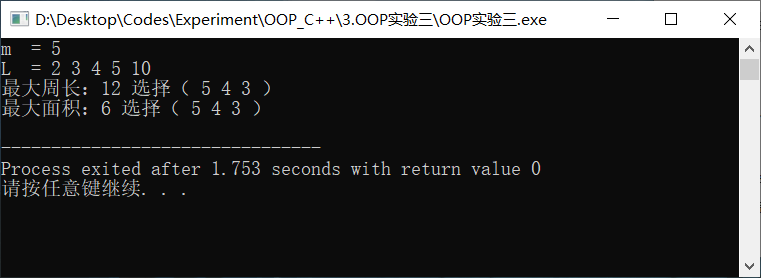
输出：最大周长12，最大面积6 (选择3, 4, 5)

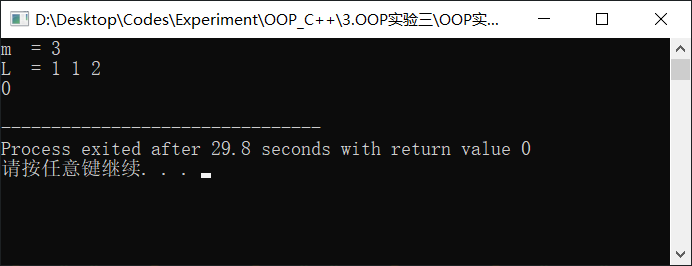
1. **实验过程**

实验环境：Dev C++

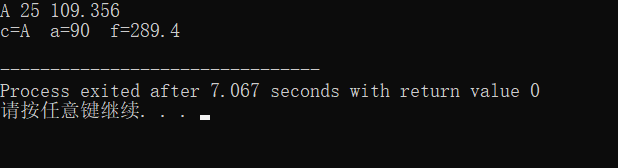
在按照题目要求格式读入数据后先对所有木棍长度从大到小进行排序，从前到后三个三个遍历木棍，易知周长最大的三角形是第一个合法构成的三角形。又可证明，在所有木棍中，面积最大的三角形其三边长度一定是长度连续的三个木棍组成，故再次遍历木棍，计算每三个木棍组成的合法三角形的面积。找到最大面积，记录最大面积的木棍编号。最后输出即可。

1. **实验结果**





（1）例2.1



分析：将输出改为如下格式



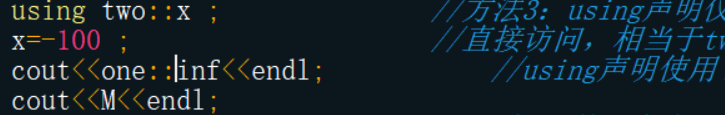
（2）例2.4

报错



分析：inf被重复定义

解决：



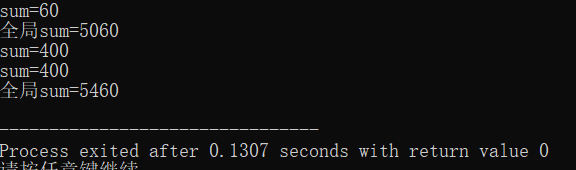
（3）例2.5

①没有变化

②Dev C++未报错（）

分析：一、变量可以随用随定义，二、IDE太智能了。

（4）例2.6

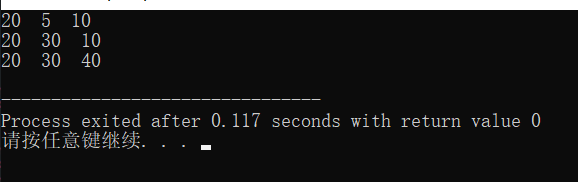


分析：int sum = 200 提前，第三个输出的sum没有使用域选择符，故输出局部变量200 \* 2的值，因为局部变量200自乘了2，故变为400，第四句输出400，因为自乘操作用在了局部变量上，则全局变量不变仍为5460.

（5）例2.7

①报错

分析：对于第一句传入参数过少（第二个参数没有设置默认值）

②

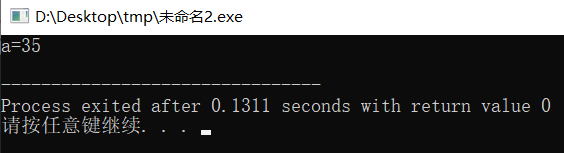
分析：与原函数输出无变化，i的默认参数并没有用到

③报错

分析：默认形参要放在最后

④同上

（6）例2.8



分析：如此更改替换时加上括号就没有问题了

（7）例2.9

报错

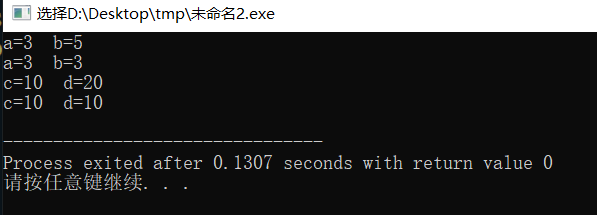
分析：多个带默认参数的函数产生了二义性

（8）例2.10

报错：

分析：引用被声明时必须被初始化

（9）例2.11



分析：因为没有对x进行传引用操作，函数不会对第一个参数即主函数中的a、c变量产生影响，但是会改变第二个参数的值，就导致第二个变量的值被改为第一个变量的值，但第一个变量的值并没有发生改变。

1. **实验总结**

scanf()不能读入空格，要用getchar()吃掉再进行下面的读入。

Int型函数可以返回一些不可能存在的数值以起到bool型的功能，同理，int型变量也可以起到存数据和bool型两个功能。

* 1. Cout精度控制可以使用setprecision实现
  2. 使用多个域时要小心重复声明，使用域选择符区分
  3. IDE可以帮你忽略掉一些错误
  4. 注意全局变量和局部变量的区分
  5. 注意函数重载时必须要规避掉二义性，默认参数要放到最后
  6. 宏定义是简单替换具有风险，可以加一些括号规避掉部分风险
  7. 引用被声明时必须被初始化
* **附录：程序源码（建议基于Highlight软件导入）**

****

1. #include<bits/stdc++.h>
2. #define ios ios::sync\_with\_stdio(false);cin.tie(0);cout.tie(0);
3. #define debug(a) cout << #a << " " << a << endl
4. **using** **namespace** std;
5. **typedef** **long** **long** ll;
6. **const** **double** pi=acos(-1);
7. **const** **double** eps = 1e-8;
8. **const** **int** inf = 0x3f3f3f3f;
9. **const** **int** maxn = 100007;//1e5+7
10. **const** ll mod = 1000000007;//1e9+7
12. **double** a[maxn];
14. **double** Area(**double** a,**double** b,**double** c)
15. {
16. **if** (a >= b + c) {
17. **return** 0;
18. }
19. **double** p = (a + b + c) \* 1.0 / 2;
20. **double** ar = sqrt(p \* (p - a) \* (p - b) \* (p - c));
21. **return** ar;
22. }

25. **bool** cmp(**double** a,**double** b)
26. {
27. **return** a > b;
28. }
30. **int** main()
31. {
32. **int** m;
33. scanf("m  = %d",&m);
34. getchar();
35. scanf("L  = ");
36. **for**(**int** i = 1;i<=m;i++){
37. cin>>a[i];
38. }
39. /\*  for(int i = 1;i<=m;i++){
40. cout<<a[i]<<" \n"[i == m];
41. }\*/
42. sort(a+1,a+m+1,cmp);
43. **int** posc = -1,poss = -1;
44. **double** mxs = 0,mxc = 0;
45. **for**(**int** i = 1;i+2<=m;i++){
46. **if**(a[i] < (a[i+1] + a[i+2])){
47. posc = i;
48. mxc = a[i] +a[i+1] + a[i+2];
49. **break**;
50. }
51. }
52. **if**(posc != -1){
53. **for**(**int** i = 2;i+1<=m;i++){
54. **double** areaa = Area(a[i-1],a[i],a[i+1]);
55. **if**(areaa > 0){
56. **if**(areaa > mxs){
57. mxs = areaa;
58. poss = i;
59. }
60. }
61. }
62. cout<<"最大周长："<< mxc<<" 选择（ "<<a[posc]<<" "<<a[posc+1]<<" "<<a[posc+2]<<" ）"<<endl;
63. cout<<"最大面积："<< mxs<<" 选择（ "<<a[poss-1]<<" "<<a[poss]<<" "<<a[poss+1]<<" ）"<<endl;
64. }
65. **else**{
66. cout<<0<<endl;
67. }
68. **return** 0;
69. }

Github地址：

<https://github.com/JiBinquan/OOP_Homework_CS2020/tree/master/3.OOP实验三>