

# 第四次作业实验报告

源代码附在压缩文件包里

## 第一题：函数计算（输出两函数的值）

### 题目

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$
$$f(x) = \frac{1}{3} \sin(x^2 + y^2) \cos(x + y)$$

### 思路

通过调用库中的sqrt、exp、sin、cos函数，并通过定义变量pi表示 $\pi$ ，然后组合成题目所需函数即可

通过定义两函数，实现同一程序输出所求两函数的值

### 代码

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4  #define pi 3.14159265358979323846264
5
6  double f1(double x);
7  double f2(double x, double y);
8
9  int main() {
10     double x, y;
11     cout << "Please input two numbers:" << endl;
12     cin >> x >> y;
13     cout << "for the first formula: f(x)=\\frac{1}{\\sqrt{2\\pi}}e^{-\\frac{x^2}{2}} \\n" << f1(x) << endl << "for the second formula: f(x)=\\frac{1}{3}\\sin{(x^2+y^2)}\\cos{(x+y)} \\n" << f2(x, y) << endl;
14     return 0;
15 }
16
17 double f1(double x) {
18     double t;
19     t = exp(-x * x / 2) * (1 / sqrt(2 * pi));
20     return t;
21 }
22
```

```

23 double f2(double x, double y) {
24     double t;
25     t = sin(x * x + y * y) * cos(x + y) / 3;
26     return t;
27 }

```

## 运行结果截图

```

Microsoft Visual Studio 调试控制台
Please input two numbers:
0 0
for the first formula: f(x)=\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}
0.398942
for the second formula: f(x)=\frac{1}{3}\sin\{x^2+y^2\}\cos\{x+y\}
0
D:\Files\课程\程设\作业\homework\Debug\homework.exe (进程 24412) 已退出, 返回代码为: 0。
按任意键关闭此窗口...

```

## 分析总结

貌似库里没有直接调用 $\pi$ 的方法，所以需要先定义 $\pi$ 的取值，可以采用define，也可以采用constant之类的方法，或者直接用double类型即可。

## 第二题：写出下列表达式的值

### 题目

```

1  !x && x!=0
2  !(x == a) && !(y == b) && 0
3  -10 < a < -5 && b == c
4  5 > 3 && 2 || 8 < 4 - !0
5  !4 < y < 5 && 5 < b < 6
6  !x || x!=0
7  3 < x < 5 || y > 3 && y < 2

```

## 思路与结果

1. 若x为真（即x!=0为真），则!x为假；若x为假（即x!=0为假），则!x为真。即!x和x!=0不能同真假。

故原式 `!x && x!=0` 值为0

2. 无论x、y、a、b为何值，最后与0（假）做与运算，结果必定为假。

故原式 `!(x == a) && !(y == b) && 0` 值为0

3. 原式左侧无论a为何值， $-10 < a$ 的结果为0或1，而0（或1） $< -5$ 的结果总为假（0），再与右侧与运算，结果为假。

故原式 `-10 < a < -5 && b == c` 值为0

4.  $5 > 3$ 为真，和2与运算为真，和右侧进行或运算，短路，永远为真。

故原式 `5 > 3 && 2 || 8 < 4 - !0` 值为1

5.  $!4$ 为假，即0； $0 < 1$ 为真，即1； $1 < 5$ 为真，得1；右侧 $5 < 1$ 为假，得0； $0 < 6$ 为真，得1；1和1与运算，为真，得1。

故原式 `!4 < y < 5 && 5 < b < 6` 值为1

6. 同理，若x为真（即 $x != 0$ 为真），则 $!x$ 为假；若x为假（即 $x != 0$ 为假），则 $!x$ 为真。则 $||$ 两侧总有一项为真，结果为真。

故原式 `!x || x != 0` 值为1

7. 无论x为何值， $3 < x$ 的结果0或1和5进行 $<$ 运算时永远为真，或运算短路，结果为真。

故原式 `3 < x < 5 || y > 3 && y < 2` 值为1

## 第三题:甄别谎言

### 思路

记说真话为1，说谎为0，用a、b、c分别记录甲乙丙三人说真话还是假话，则可以把甲乙丙的话变为下面的形式：

甲: `b == 0`

乙: `c == 0`

丙: `b == 0`

所以可知甲丙两人同真假，甲乙、乙丙中都是一真一假，并且有且仅有1人说谎，数学上已经解决本问题。

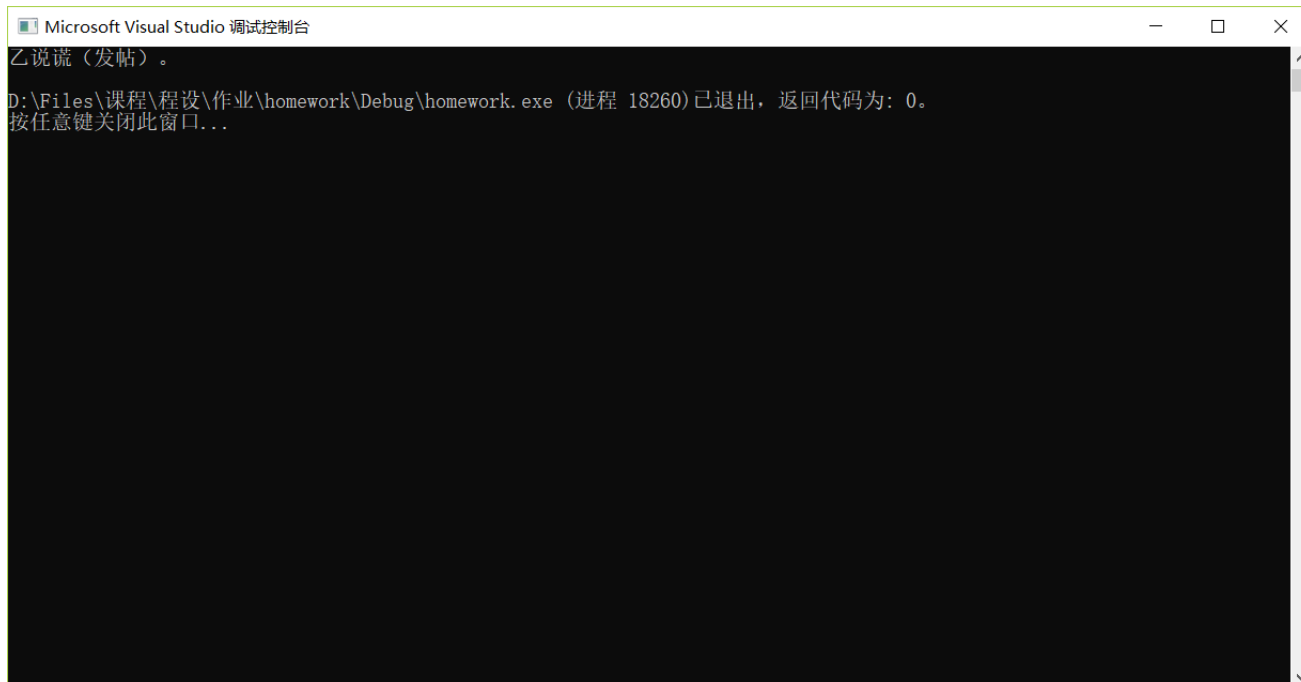
可以采取遍历每一种情况（010、101）后直接输出 $a+b+c==2$ （符合题意仅一人说谎）的情况，最简单易想也最繁琐的遍历操作如下方代码所示。

### 代码

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a, b, c;
6      // 不妨用abc来表示甲乙丙三人说话真假状态 值为1表示说真话，为0表示说假话
7      for (a = 0; a <= 1; a++) {
8          b = 1 - a;
9          c = a;
10         if (a + b + c == 2) {
11             if (a == 0) {
12                 cout << "甲说谎 (发帖) 。\n";
13                 if (b == 0) {
```

```
14         cout << "乙说谎（发帖）。\n";
15         if (c == 0) {
16             cout << "丙说谎（发帖）。\n";
17         }
18     }
19 }
20 else {
21     if (b == 0) {
22         cout << "乙说谎（发帖）。\n";
23         if (c == 0) {
24             cout << "丙说谎（发帖）。\n";
25         }
26     }
27     else {
28         if (c == 0) {
29             cout << "丙说谎（发帖）。\n";
30         }
31     }
32 }
33 }
34 }
35 return 0;
36
37 }
```

## 运行结果截图



## 分析总结

这道题相比于课堂上所讲较为简单，通过简单的逻辑分析即可得出答案。由于需要用程序实现，一个基本的想法是遍历所有情况并输出符合条件的答案，上方代码就是一个比较直接的遍历操作。

由于逻辑推理知道甲丙同真假，当甲为假的时候丙为假从而条件不成立，进一步缩短程序判断的时间。

## 选做题：买鸡

### 思路

从小鸡公鸡各一只，母鸡98只开始，可采用3层for循环遍历所有的情况，输出符合题意的情况即可

采用一个变量记录所得的最大母鸡数，之后所有比这个数目小的都排除

### 代码

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  int main() {
6      int iChicken, iHen, iRooster, max = 1;
7      cout << "所求购买方案为: \n";
8      for (iRooster = 1; iRooster < 99; iRooster++) {
9          for (iChicken = 1; iChicken < 99; iChicken++) {
10             for (iHen = 97; iHen > 1; iHen--) {
11                 if ((iChicken + iHen + iRooster == 100) && (5 * iChicken + 15 *
12 iHen + 10 * iRooster < 1001) && iHen >= max) {
13                     max = iHen;
14                     cout << "母鸡: " << iHen << "只\n公鸡: " << iRooster << "只\n小
15 鸡: " << iChicken << "只\n剩余金额: " << 1000 - (5 * iChicken + 15 * iHen + 10 *
16 iRooster) << "\n\n";
17                 }
18             }
19         }
20     }
21     return 0;
22 }
```

### 运行结果截图

```
Microsoft Visual Studio 调试控制台
所求购买方案为:
母鸡: 49只
公鸡: 1只
小鸡: 50只
剩余金额: 5

母鸡: 49只
公鸡: 2只
小鸡: 49只
剩余金额: 0

D:\Files\课程\程设\作业\homework\Debug\homework.exe (进程 27184) 已退出, 返回代码为: 0。
按任意键关闭此窗口...
```

## 分析总结

由于母鸡所用变量在循环中递减, 故第一次满足情况时母鸡数量49肯定为所有情况中最大的, 事实上将所有符合金额条件的方案全部输出到txt文件中进行确认后49确为所求数量。通过定义max变量表示最大母鸡数并于第一次符合所有条件时赋值当时的母鸡数量, 可以确保后续遍历操作中不输出母鸡数量少于max的。

三个for循环遍历在逻辑上很直接, 但是有很多不必要的操作, 占用了大量时间和空间。更好的操作可能为将小鸡和公鸡的数量放在同一循环中, 循环条件为金钱进行遍历操作。设想大致如下方伪代码:

```
1  int max = 1;
2  for(iHen = 97; iHen > max; iHen--){
3      for(iRooster = 1, iChicken = 100 - iHen - iRooster; 5 * iChicken + 15 * iHen + 10
        * iRooster < 1001; iRooster++){
4          max = iHen;
5          cout << 方案情况;
6      }
7  }
```

这种设想可能可以减少时间复杂度和内存占用量, 实际情况尚未检验。