

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级： 网安2003班**

**学 号： U202012068**

**姓 名： 叶礼亮**

**指导教师： 余林琛**

**报告日期： 2020/10/24**

**网络空间安全学院**

**目 录**

[**1 表达式和标准输入输出实验 1**](#_Toc404837920)

[1.1 实验目的 1](#_Toc404837921)

[1.2 实验内容 1](#_Toc404837922)

[1.3 实验小结 1](#_Toc404837923)

[**2 流程控制实验 2**](#_Toc404837924)

[2.1 实验目的 1](#_Toc404837921)

[2.2 实验内容 1](#_Toc404837922)

[2.3 实验小结 1](#_Toc404837923)

[**3 函数与程序结构实验 3**](#_Toc404837929)

[3.1 实验目的 1](#_Toc404837921)

[3.2 实验内容 1](#_Toc404837922)

[3.3 实验小结 1](#_Toc404837923)

[**4 编译预处理实验 4**](#_Toc404837934)

[4.1 实验目的 1](#_Toc404837921)

[4.2 实验内容 1](#_Toc404837922)

[4.3 实验小结 1](#_Toc404837923)

[**5 数组实验 5**](#_Toc404837938)

[5.1 实验目的 1](#_Toc404837921)

[5.2 实验内容 1](#_Toc404837922)

[5.3 实验小结 1](#_Toc404837923)

[**6 指针实验 6**](#_Toc404837943)

[6.1 实验目的 1](#_Toc404837921)

[6.2 实验内容 1](#_Toc404837922)

[6.3 实验小结 1](#_Toc404837923)

[**7 结构与联合实验 7**](#_Toc404837948)

[7.1 实验目的 1](#_Toc404837921)

[7.2 实验内容 1](#_Toc404837922)

[7.3 实验小结 1](#_Toc404837923)

[**8 文件实验 8**](#_Toc404837953)

[8.1 实验目的 1](#_Toc404837921)

[8.2 实验内容 1](#_Toc404837922)

[8.3 实验小结 1](#_Toc404837923)

[**参考文献 9**](#_Toc404837957)

# 2 流程控制实验

## 2.1 实验目的

（1）掌握复合语句、if语句、switch语句的使用，熟练掌握for、while、do-while三种基本的循环控制语句的使用，掌握重复循环技术，了解转移语句与标号语句。

（2）练习循环结构for、while、do-while语句的使用。

（3）练习转移语句和标号语句的使用。

（4）使用集成开发环境中的调试功能：单步执行、设置断点、观察变量值。

## 2.2 实验内容

**2.2.1 程序改错**

下面的实验2-1程序是合数判断器（合数指自然数中除了能被1和本身整除外，还能被其它数整除的数），在该源程序中存在若干语法和逻辑错误。要求对该程序进行调试修改，使之能够正确完成指定任务。

/\* 实验2-1改错题程序：合数判断器\*/

1 #include <stdio.h>

2 int main( )

3 {

4 int i, x, k, flag = 0;

5 printf("本程序判断合数，请输入大于1的整数，以Ctrl+Z结束\n");

6 while (scanf("%d", &x) !=EOF) {

7 for(i=2,k=x>>1;i<=k;i++)

8 if (!x%i) {

9 flag = 1;

10 break;

11 }

12 if(flag=1) printf("%d是合数", x);

13 else printf("%d不是合数", x);

14 }

15 return 0;

16 }

**解答：**

（1）错误修改：

1) 第8行的判断内运算优先级有错，正确形式为：

if (!(x%i)) {

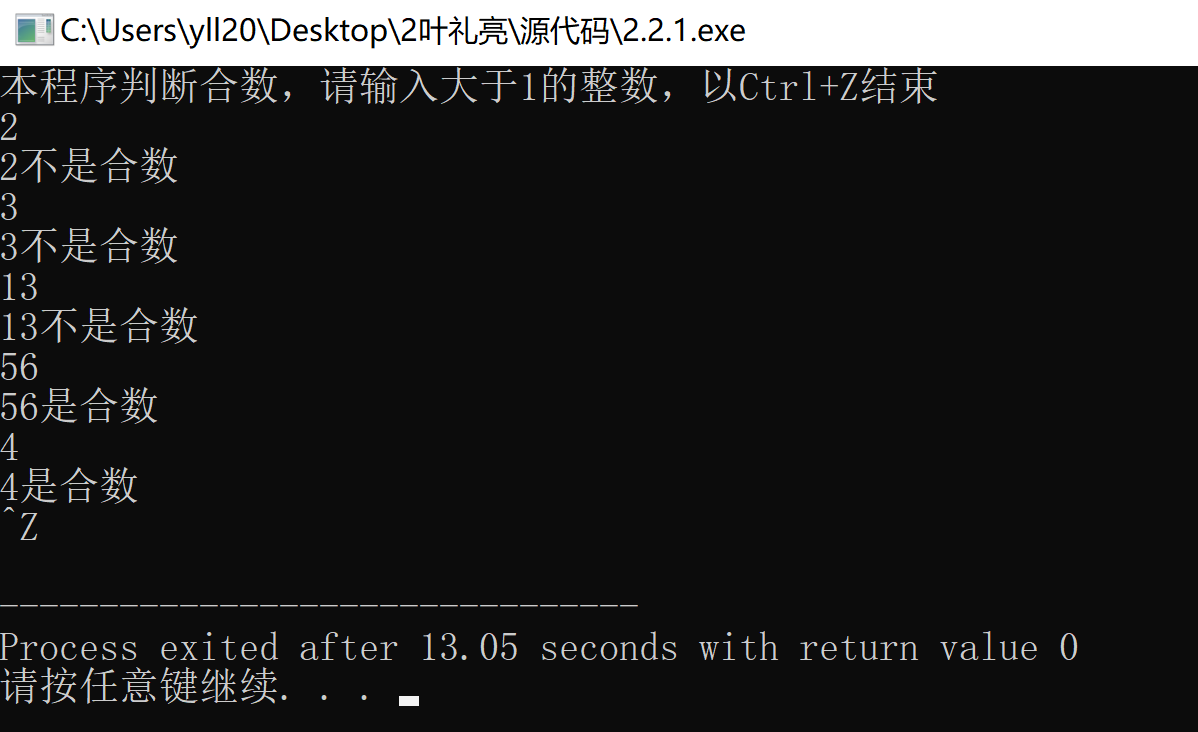
2) 第12行的判断相等应该用==，正确形式为：

if(flag==1) printf("%d是合数", x);

3) flag每次循环都应该复位，正确形式为：

添加flag = 0;

（2）错误修改后运行结果：



**2.2.2 源程序修改替换**

（1）修改实验2-1程序，将内层两出口的for循环结构改用单出口结构，即不允许使用break、goto等非结构化语句。

**解答：**

修改后的程序如下所示：

/\* 实验2-1改错题程序：合数判断器\*/

#include <stdio.h>

int main()

{

int i, x, k, flag = 0;

printf("本程序判断合数，请输入大于1的整数，以Ctrl+Z结束\n");

while (scanf("%d", &x) !=EOF)

{

for(i=2,k=x>>1;i<=k&&flag==0;i++)//加了一句&&flag==0

{

if (!(x%i))

{

flag = 1;

}

}

if(flag==1)

{

printf("%d是合数", x);

}

else

{

printf("%d不是合数", x);

}

flag = 0;

}

return 0;

}



（2）修改实验2-1程序，将for循环改用do-while循环。

修改后的程序如下所示:

/\* 实验2-1改错题程序：合数判断器\*/

#include <stdio.h>

#include<iostream>

using namespace std;

int main( )

{

int i, x, k, flag = 0;

printf("本程序判断合数，请输入大于1的整数，以Ctrl+Z结束\n");

while (scanf("%d", &x) !=EOF)

{

i=2,k=x>>1;

do

{

if (i>k) break;

if (!(x%i))

{

flag = 1;

}

i++;

}while(i<=k&&flag==0);

if(flag==1)

{

printf("%d是合数", x);

}

else

{

printf("%d不是合数", x);

}

flag = 0;

}

return 0;

}



（3）修改实验2-1程序，将其改为纯粹合数求解器，求出所有的3位纯粹合数。一个合数去掉最低位，剩下的数仍是合数；再去掉剩下的数的最低位，余留下来的数还是合数，这样反复，一直到最后剩下一位数仍是合数，这样的数称为纯粹合数。

修改后的程序如下所示:

#include <stdio.h>

int unit\_num(int n)

{

int i;

for(i=2;i<=n>>1;i++)

{

if(n%i==0)

return 1;

}

return 0;

}

int main()

{

int i,a,b;

for(i=100;i<1000;i++)

{

a=i/100;

b=i/10;

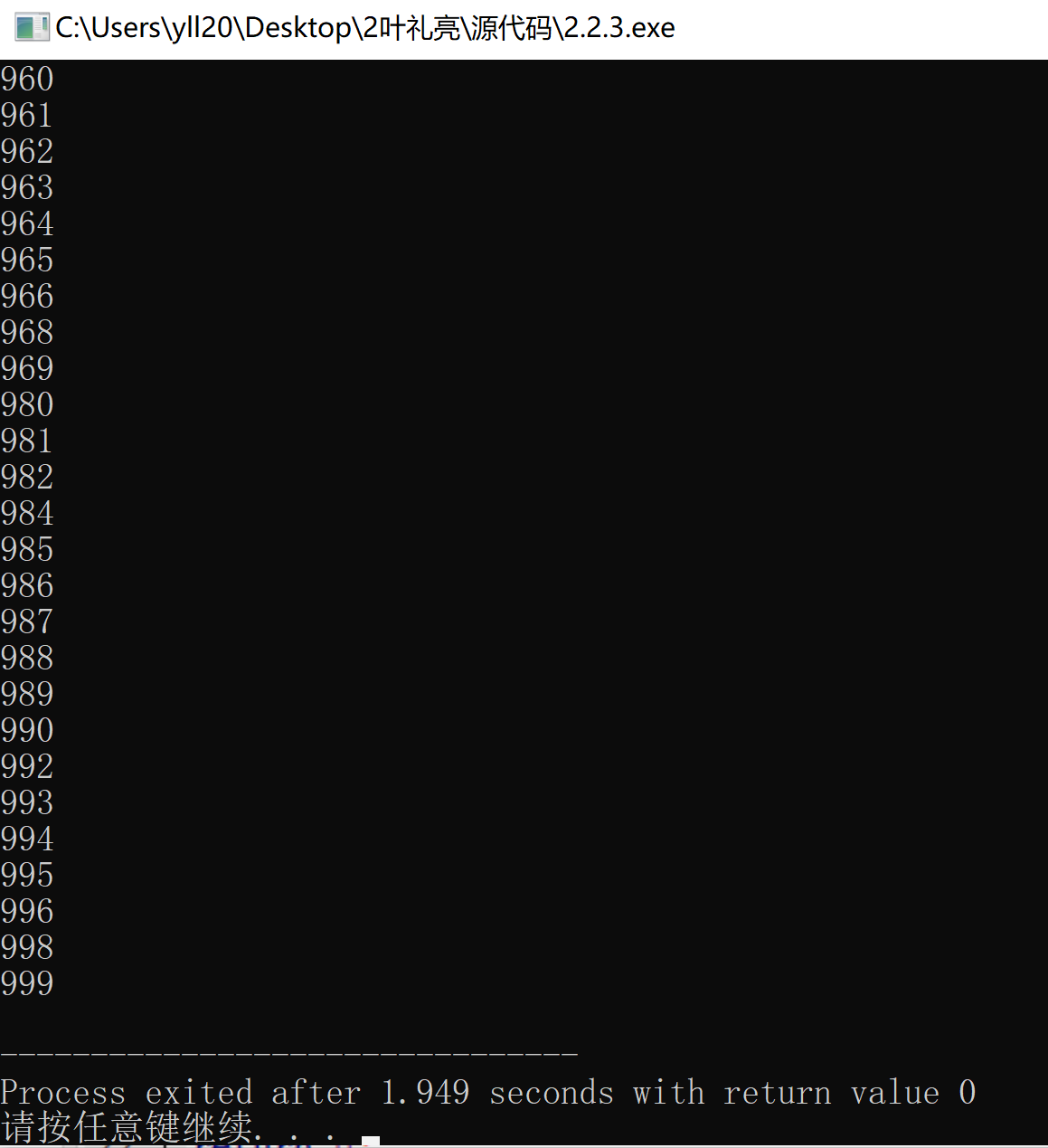
if(unit\_num(i)&&unit\_num(a)&&unit\_num(b)&&a>1)

printf("%d\n",i);

}

return 0;

}



**2.2.3 程序设计**

（1） 假设工资税金按以下方法计算：x ＜ 1000元，不收取税金；1000 ≤ x ＜ 2000，收取5%的税金；2000 ≤ x ＜ 3000，收取10%的税金；3000 ≤ x ＜ 4000，收取15%的税金；4000 ≤ x ＜ 5000，收取20%的税金；x＞5000，收取25%的税金。（注意税金的计算按照阶梯计税法，比如，工资为4500，那么税金=1000\*5% + 1000\*10% + 1000\*15% + 501\*20%）。编写一个程序,输入工资金额，输出应收取税金额度，要求分别用if语句和switch语句来实现。

**解答：**

1. 算法流程如图1.1所示。

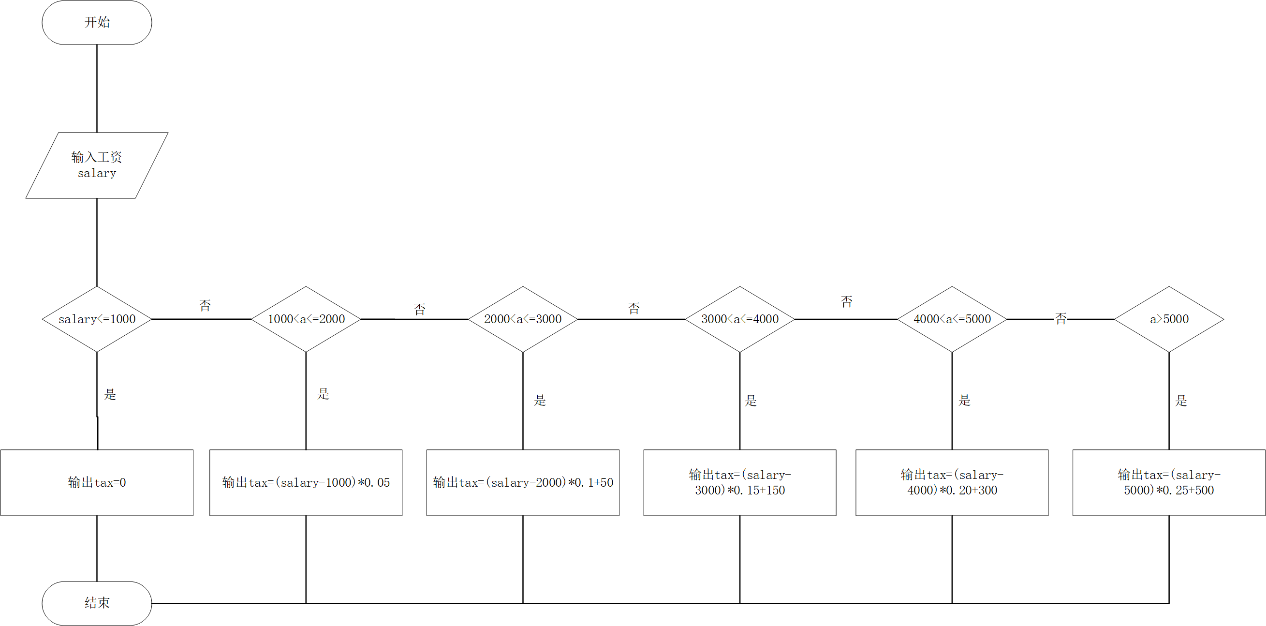


图1-1 编程题1的程序流程图

2）源程序清单

#include <stdio.h>

int main()

{

int salary, tax;

int a;

scanf("%d",&salary);

a=salary/1000;

if(a==0) tax=0;

else if(a==1) tax=(salary-1000)\*0.05;

else if(a==2) tax=(salary-2000)\*0.1+50;

else if(a==3) tax=(salary-3000)\*0.15+150;

else if(a==4) tax=(salary-4000)\*0.20+300;

else tax=(salary-5000)\*0.25+500;

printf("%d",tax);

return 0;

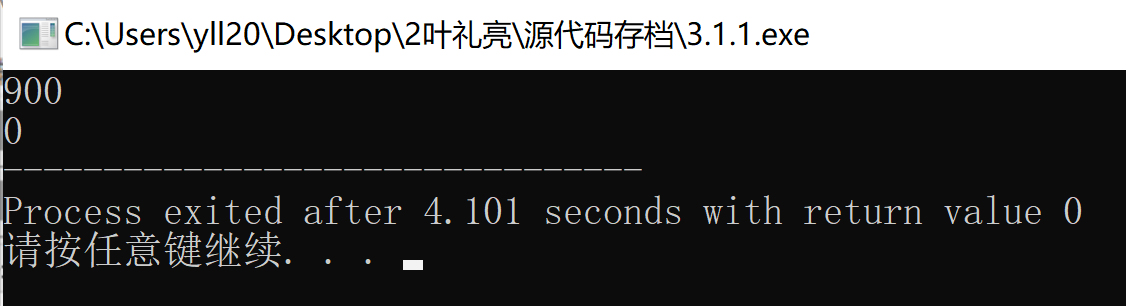
}

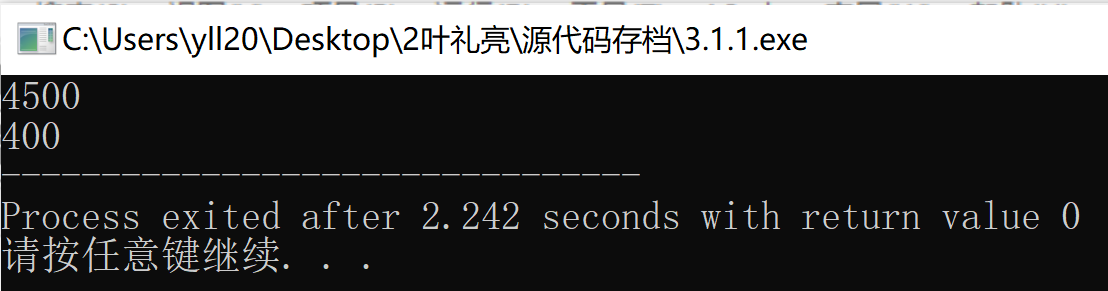
3）测试

（a） 测试数据：

900,4500

（b） 对应测试数据的运行结果截图





（2）将输入的正文复制到输出，复制过程中将每行一个以上的空格字符用一个空格代替。

1） 算法流程如图1.1所示。

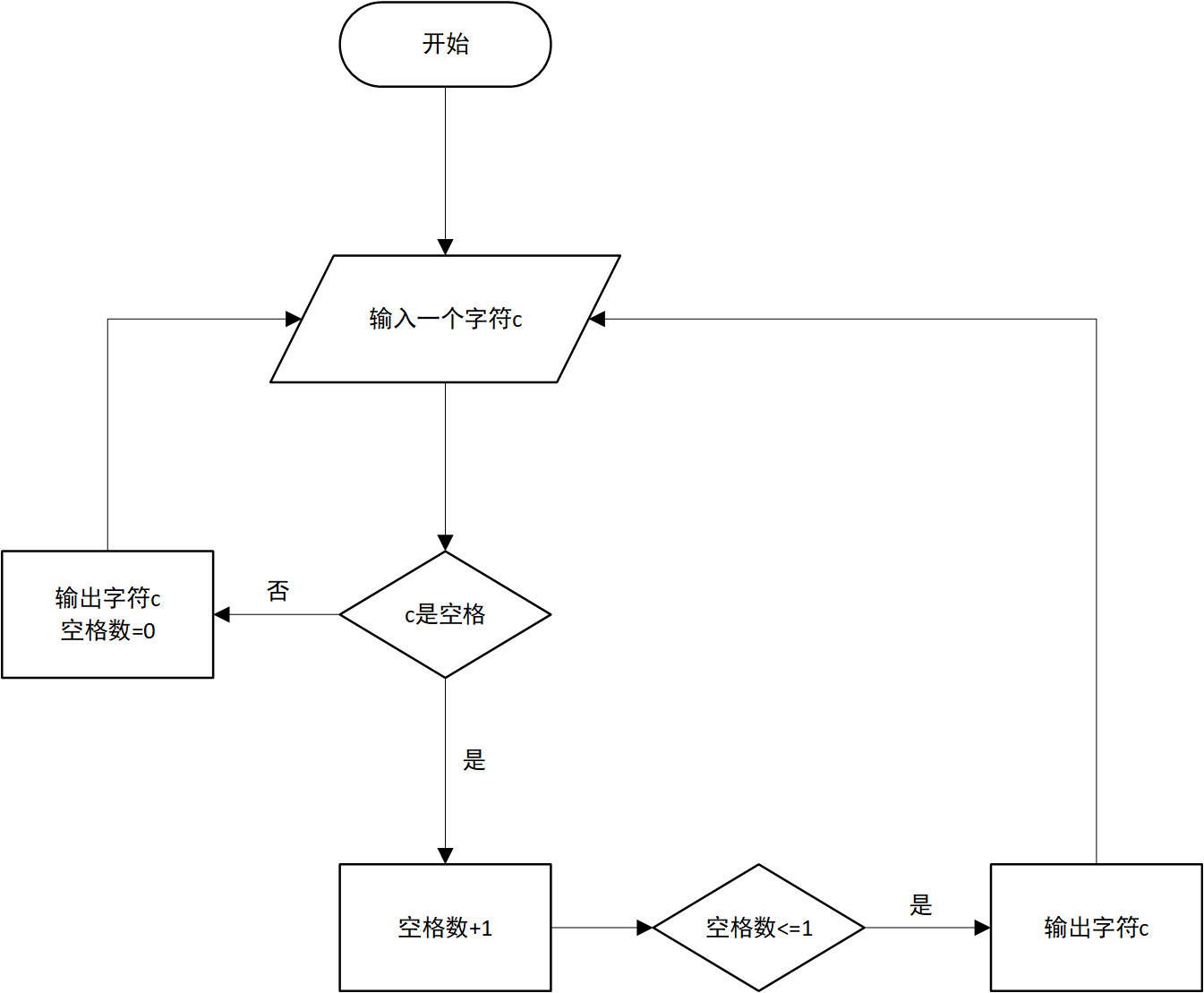


图2-1 编程题2的程序流程图

2）源程序清单

#include <stdio.h>

int main()

{

int c, space\_num;

space\_num=0;

while((c=getchar())!=EOF)

{

if (c!=' ')

{

putchar(c);

space\_num=0;

}

else

{

space\_num++;

if (space\_num<=1) putchar(c);

}

}

return 0;

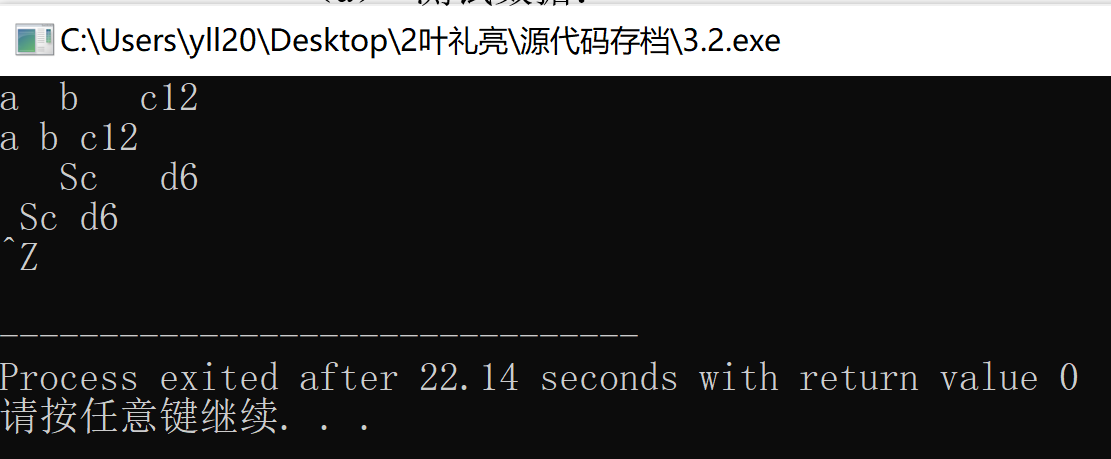
}

3）测试

（a） 测试数据：

1.a b c12 2. Sc d6

（b） 对应测试数据的运行结果截图



（3）打印如下的杨辉三角形。

1 /\*第0行 \*/

1 1 /\*第1行 \*/

1 2 1 /\*第2行 \*/

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

1 8 28 56 70 56 28 8 1

1 9 36 84 126 126 84 36 9 1

第i行第j列位置的数据值可以由组合表示，而的计算如下：

|  |  |
| --- | --- |
|  | (i=0,1,2,…) |
|  | (j=0,1,2,3,…,i) |

根据以上公式，采用顺推法编程，输出金字塔效果的杨辉三角形。特别要注意空格的数目，一位数之间是3个空格，两位数之间有2个空格，3位数之间只有一个空格。

1. 算法流程如下所示。
2. 输入需要输出的行数
3. 每一层开头输出行数减去几个数字+1
4. 调用获取值的函数,四格向左对齐输出
5. 源程序清单

#include<stdio.h>

int number(int i, int j);

int main(void)

{

int i, j, m;

int howmany\_lines;

scanf("%d",&howmany\_lines);

for(i = 0; i < howmany\_lines; i++)

{

for(m = howmany\_lines-i-1; m > 0; m--)

printf(" ");

for(j = 0; j <= i; j++)

printf("%-4d", number(i, j));

printf("\n");

}

return 0;

}

int number(int i, int j)

{

int sum;

if (j==0)

sum = 1;

else

sum = number(i, j-1)\*(i - j + 1)/j;

return sum;

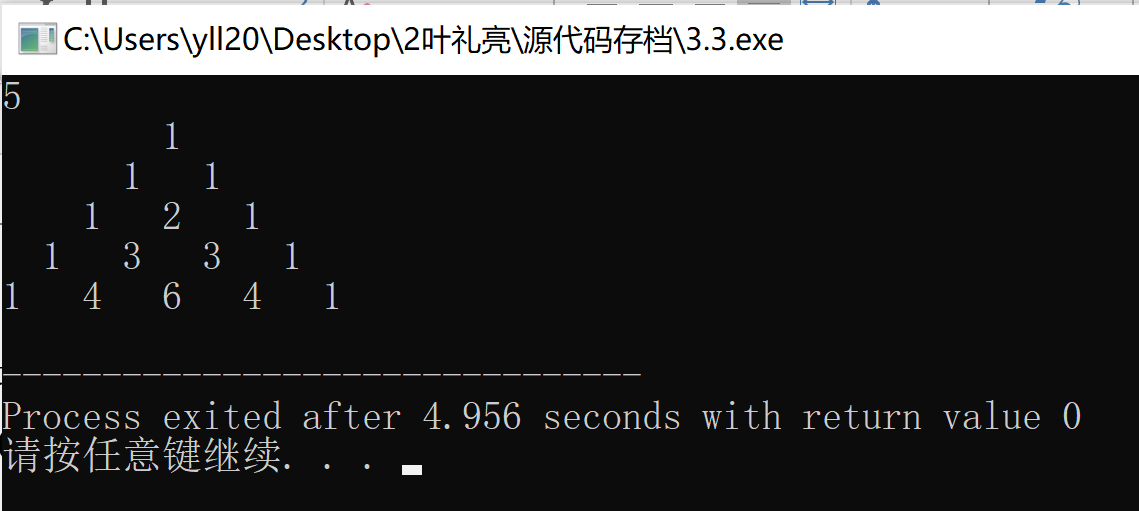
}

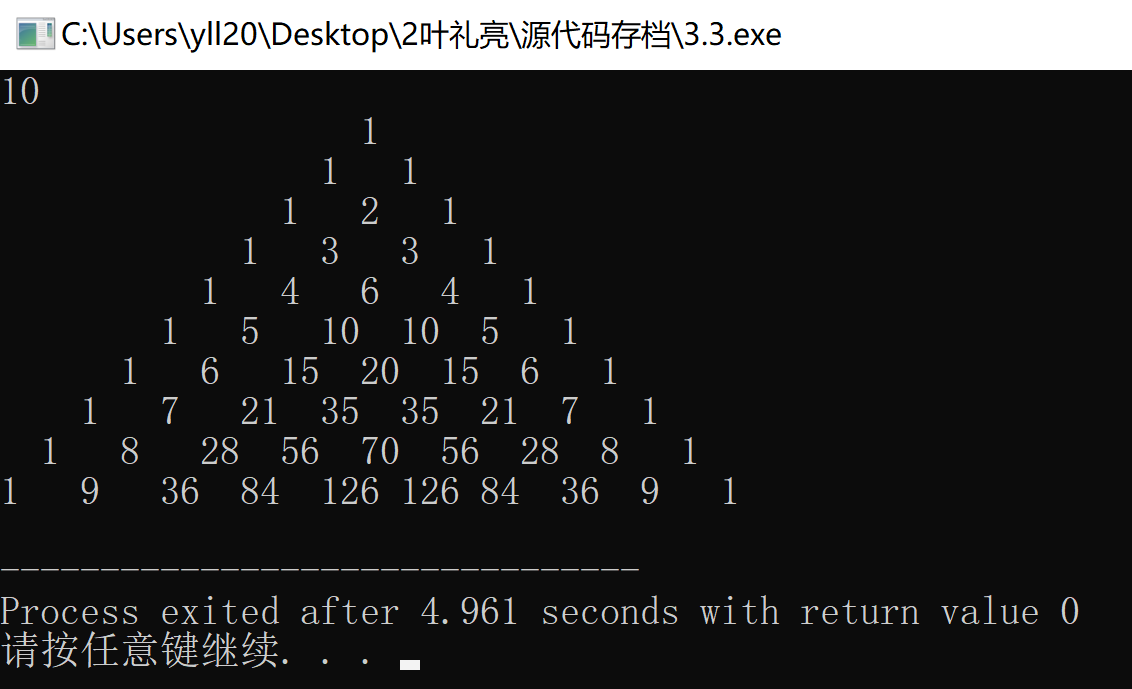
3）测试

（a） 测试数据：

5,10

（b） 对应测试数据的运行结果截图





（4）625这个数很特别，625的平方等于390625，其末3位也是625。请编程输出所有这样的3位数：它的平方的末3位是这个数本身。

1） 算法流程如图4.1所示。

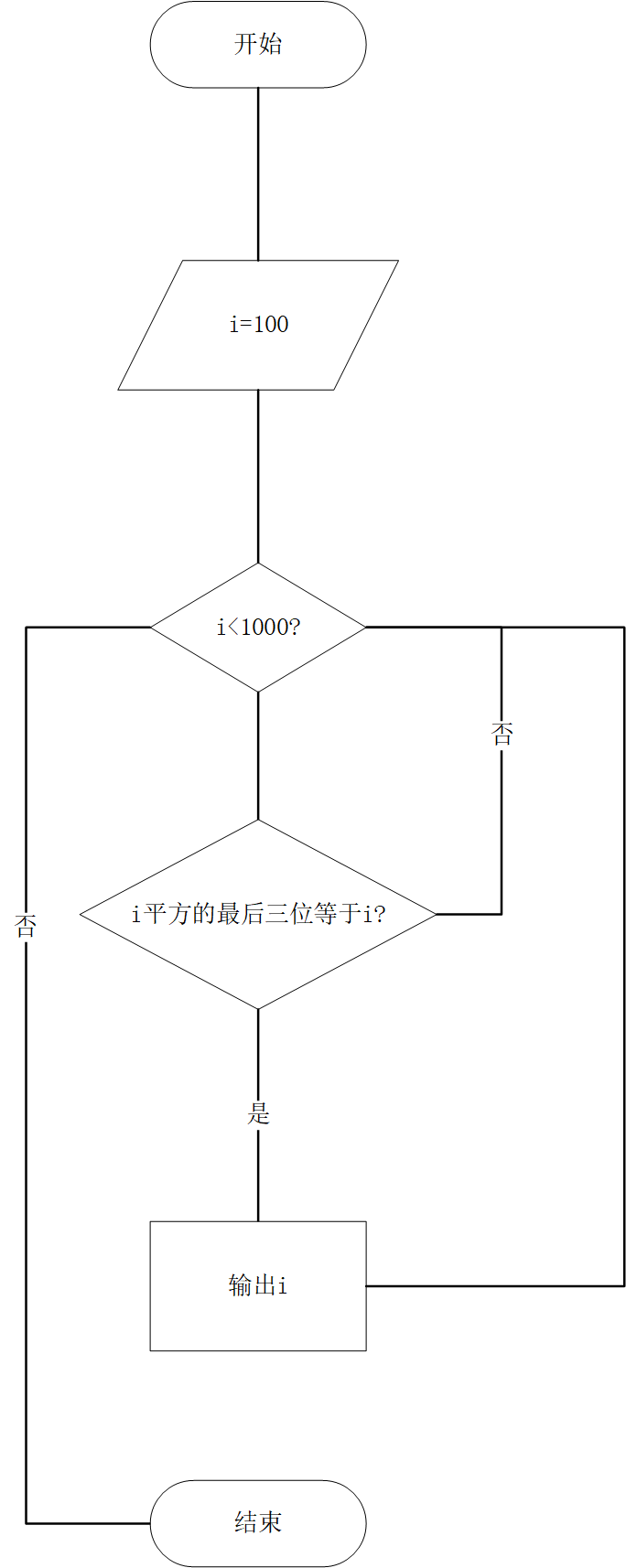


图4-1 编程题4的程序流程图

2）源程序清单

#include<stdio.h>

int main()

{

int i, i\_2, last\_3;

for (i=100;i<1000;i++)

{

i\_2=i\*i;

last\_3=i\_2%1000;

if(last\_3==i)

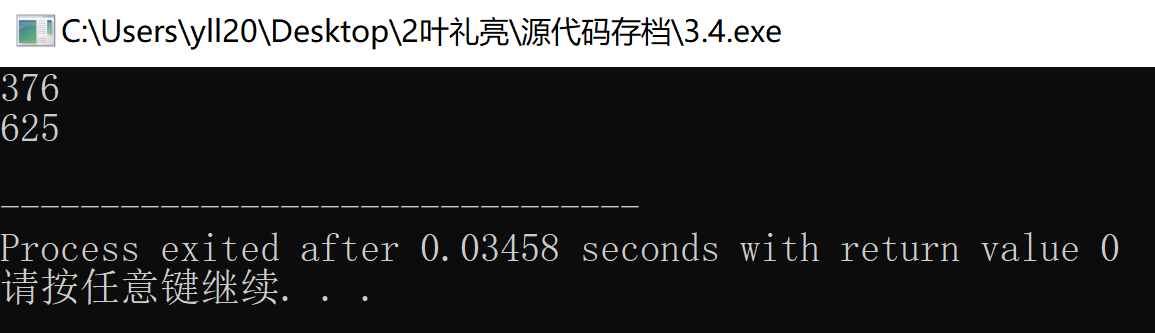
printf("%d\n",i);

}

return 0;

}

3）测试



## 1.3 实验小结

实验中很多地方的算法有累赘或者效率低的地方,修修补补才弄好.

以后要想得更周到再写代码.

参考文献

[1] 曹计昌,卢萍,李开. C语言程序设计,北京： 科学出版社,2013

[2] 李开,卢萍,曹计昌. C语言实验与课程设计, 北京：科学出版社,2011