

**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： U202012043**

**姓 名： 范启航**

**指导教师： 张云鹤**

**报告日期： 2020年10月24日**

**网络空间安全学院**

**目□□录**

**1**[**□□□表达式和标准 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\*/

#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;i++)

if (!x%i) {

flag = 1;

break;

}

if(flag=1) printf("%d是合数", x);

else printf("%d不是合数", x);

}

return 0;

**解答：**

（1）错误修改：

1）第8行中if内少了括号，应为if(!(x%i)

2）第12行中if语句中判断是否相等应用==号，改为if(flag == 1)

3）while循环中flag应复位为0；while循环中插入flag = 0;

（2）错误修改后运行结果：

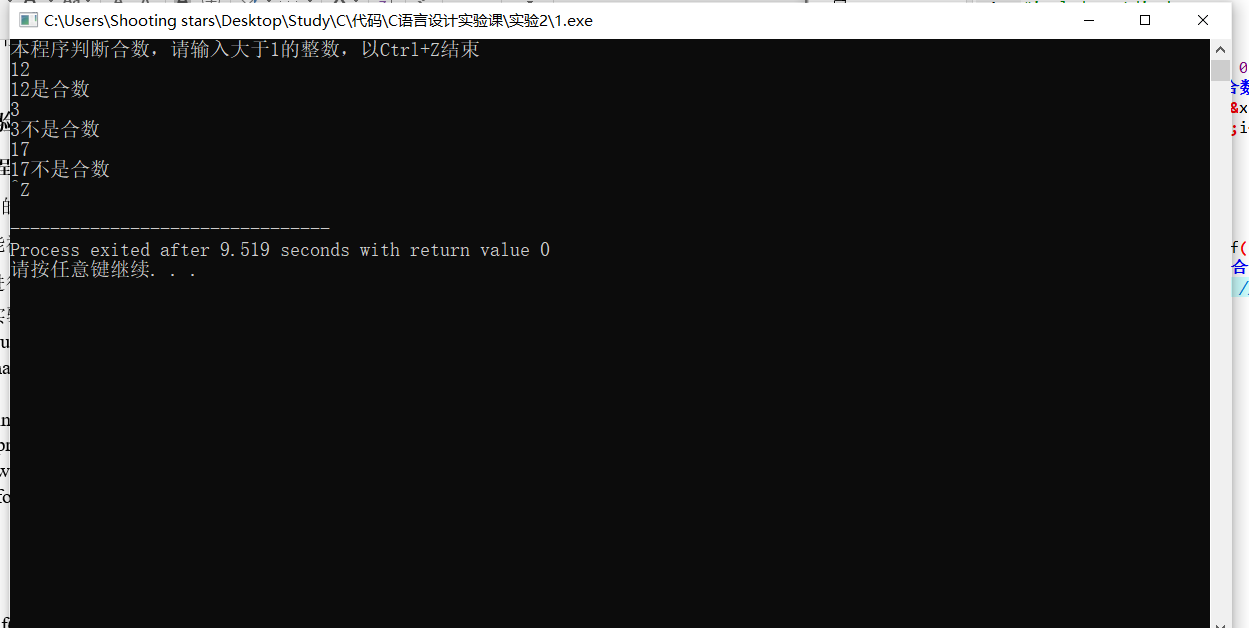


图2-2-1 运行结果示意图

**2.2.2．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。

（2）修改实验2-1程序，将for循环改用do-while循环。

（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。

**解答：**

（1）（2）替换后的如下：

#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) {

if (!(x==2))

{

i = 2, k = x>>1;

do{

if (!(x%i))

{

flag = 1;

i = k;

}

i ++;

}while(i <= k);

}else

flag = 0;

if(flag == 1) printf("%d是合数", x);

else printf("%d不是合数", x);

flag = 0;

}

return ；

运行结果示意图：

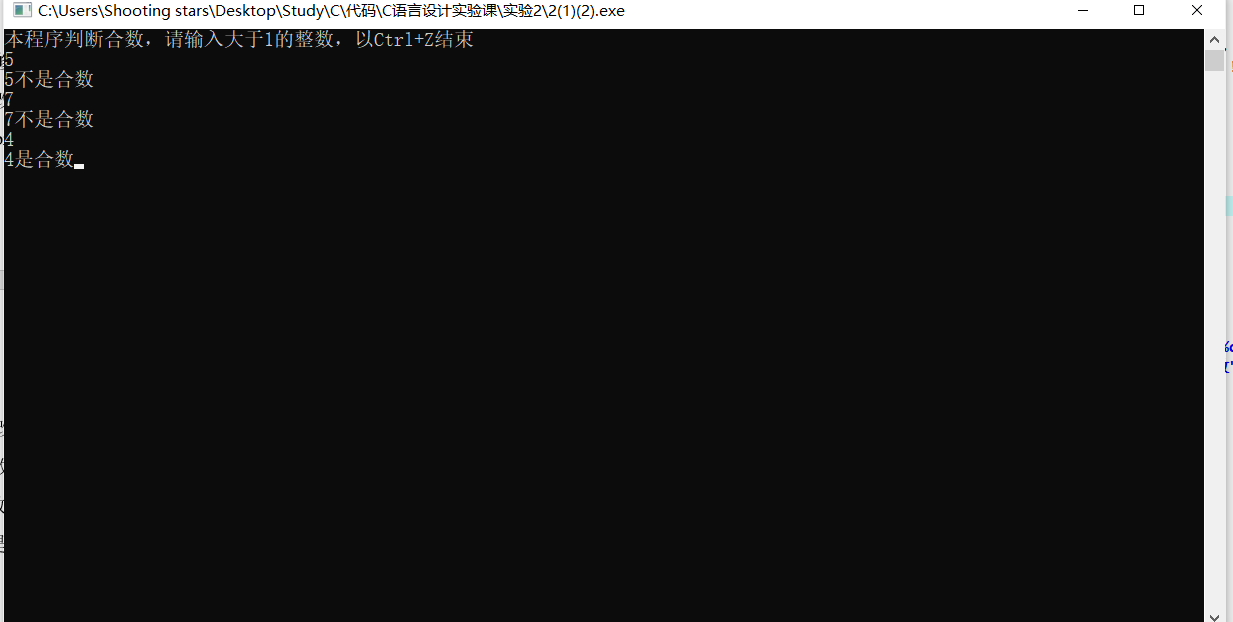


图2-2-2运行结果示意图

（3）代码如下：

#include<stdio.h>

int isheshu(int );

int main()

{

int heshu3,flag = 0;

int heshu;

for(heshu3 = 100;heshu3<=999;heshu3++)

{

heshu = heshu3;

if(isheshu(heshu))

{

if(isheshu(heshu/10))

{

heshu /=10;

if(isheshu(heshu/10))

printf("%d是纯粹合数\n", heshu3);

}

}

}

return 0;

}

int isheshu(int x )

{

int i,k,flag;

i = 2, k = x>>1, flag = 0;

if(!(x ==0 || x==1|| x==2))

{

do{

if (!(x%i))

{

flag = 1;

i = k;

}

i ++;

}while(i <= k);

}

else flag = 0;

return flag;

}

运行示例图：

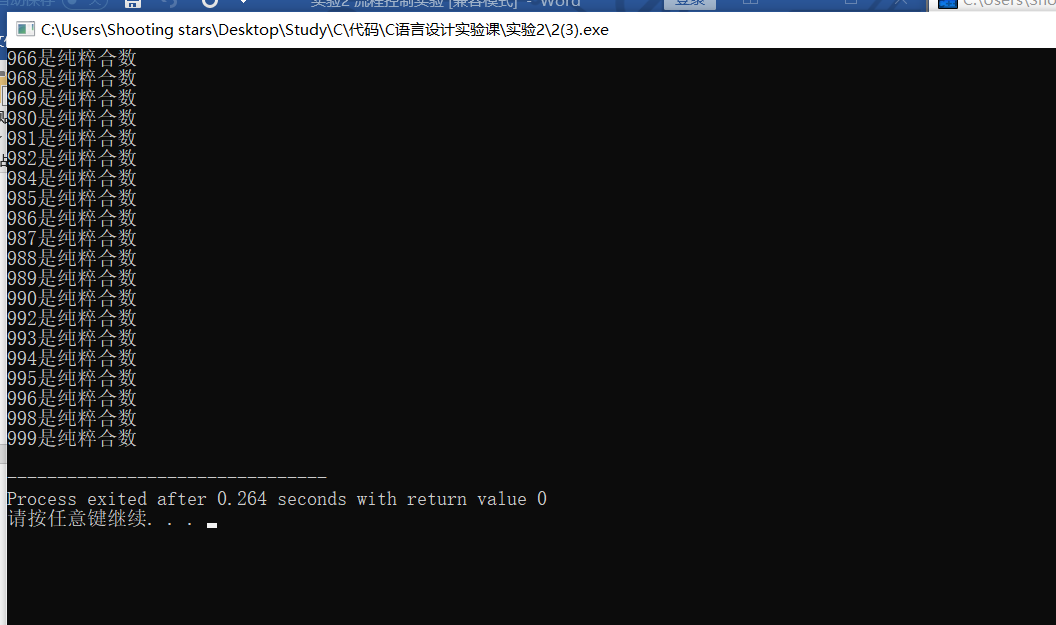


图2-2-3运行结果示意图

**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：

流程图如下：



图2-2-4 程序设计流程图

代码如下：

switch语句：

#include <stdio.h>

int main()

{

int x;

int n =0;

scanf("%d", &x);

switch(x/1000)

{

case 0: n=0;

break;

case 1: n=(x-1000)\*0.05;

break;

case 2: n=1000\*0.05+(x-2000)\*0.1;

break;

case 3: n=1000\*0.05+1000\*0.1+(x-3000)\*0.15;

break;

case 4: n=1000\*0.05+1000\*0.1+1000\*0.15+(x-4000)\*0.2;

break;

default : n=1000\*0.05+1000\*0.1+1000\*0.15+1000\*0.2+(x-5000)\*0.25;

break;

}

printf("%d", n);

return 0;

}

if语句：

#include<stdio.h>

int main()

{

int x;

int n=0;

scanf("%d", &x);

if(x>0 && x<1000) n=0;

else if(x>=1000 && x<2000) n=(x-1000)\*0.05;

else if(x>=2000 && x<3000) n=1000\*0.05+(x-2000)\*0.1;

else if(x>=3000 && x<4000) n=1000\*0.05+1000\*0.1+(x-3000)\*0.15;

else if(x>=4000 && x<5000) n=1000\*0.05+1000\*0.1+1000\*0.15+(x-4000)\*0.2;

else if(x>=5000) n=1000\*0.05+1000\*0.1+1000\*0.15+1000\*0.2+(x-5000)\*0.25;

printf("%d", n);

return 0;

}

运行示例：

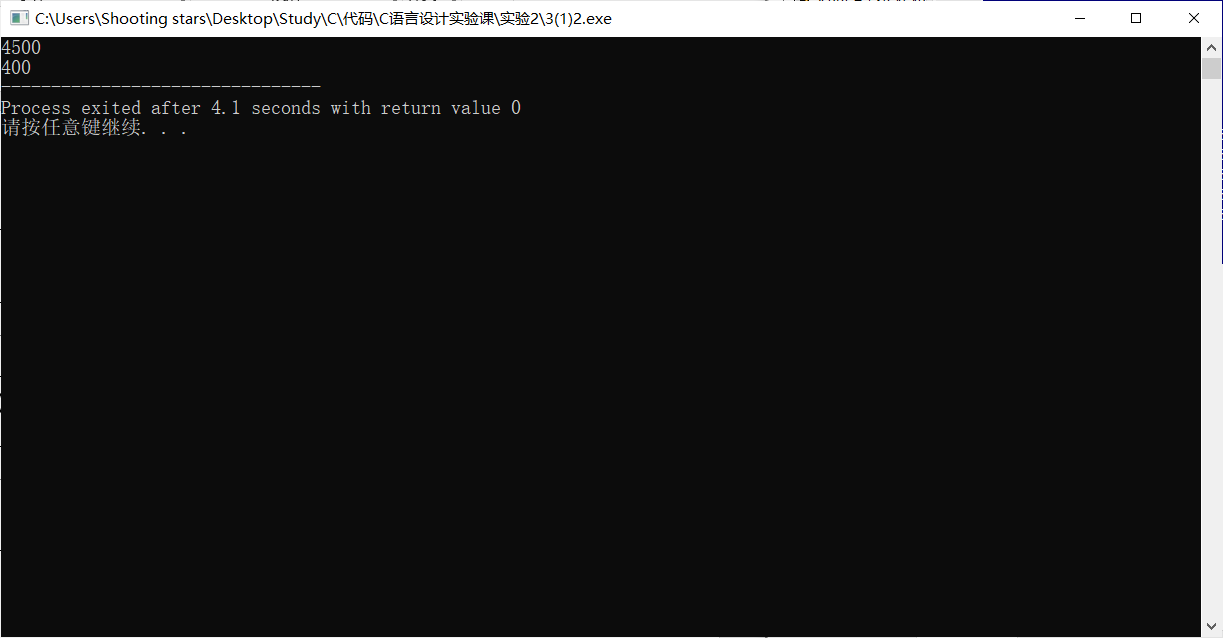


图2-2-5 运行结果示意图

（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。

解答：

流程图：



图2-2-6 程序设计流程图

代码：

#include<stdio.h>

int main()

{

char c,is;

is = 1;

while((c = getchar()) !='\n')

{

if(c == ' ')

{

if(is)

{ putchar(c);

is = 0;

}

}

else

{

putchar(c);

is =1;

}

}

return 0;

}

运行示例 ：

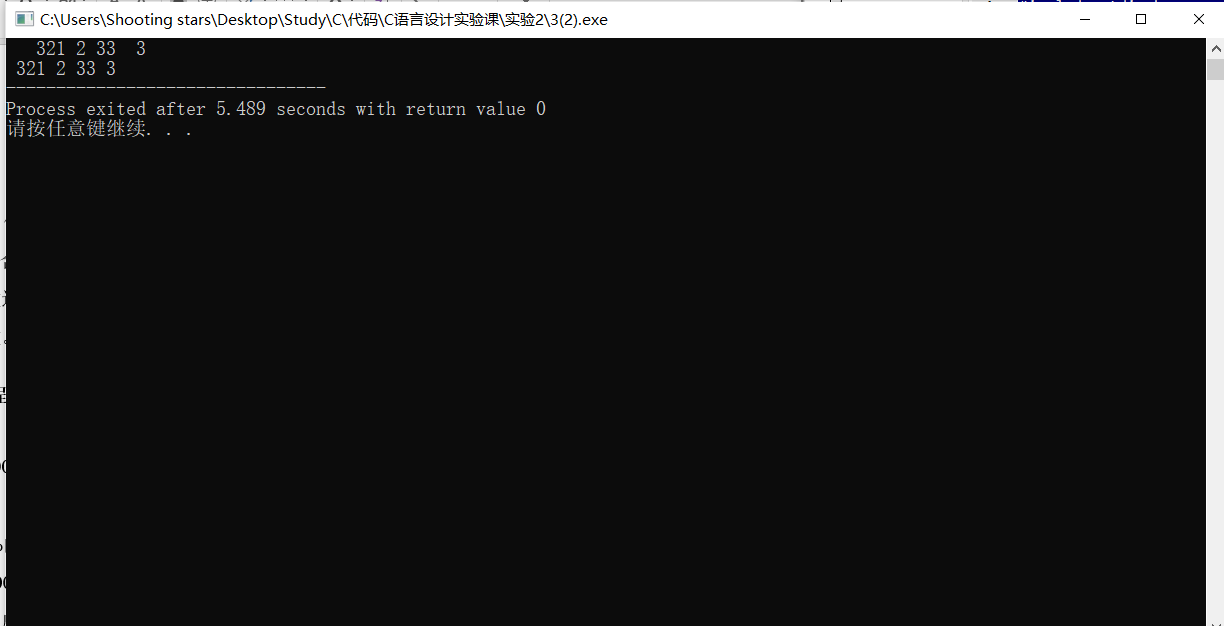


图2-2-7 运行结果示意图

（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。

解答：

代码：

#include <stdio.h>

int main()

{

int number;

scanf("%d", &number);

int a[number+1][number+1];

int i,j,k;

k = 0;

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

{

a[i][0] = 1;

}

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

{

for(j = 1; j<=i; j++)

{

a[i][j]= a[i][j-1]\*(i-j+1)/j;

}

}

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

{

while(k<2\*number-2\*i)

{

printf(" ");

k++;

}

k = 0;

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

{

printf("%d", a[i][j]);

if(a[i][j] /100)

printf(" ");

else if(a[i][j]/10)

printf(" ");

else printf(" ");

}

if(i<=number)

printf("\n");

}

return 0;

}

运行示例：



图2-2-8 运行结果示意图

（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。、

解答：

流程图：



图2-2-9程序设计流程图

代码：

#include<stdio.h>

int main()

{

int i,j;

for(i = 100; i<=999; i++)

{

j = i\*i;

if(i == j%1000)

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

}

return 0;

}

运行示例：

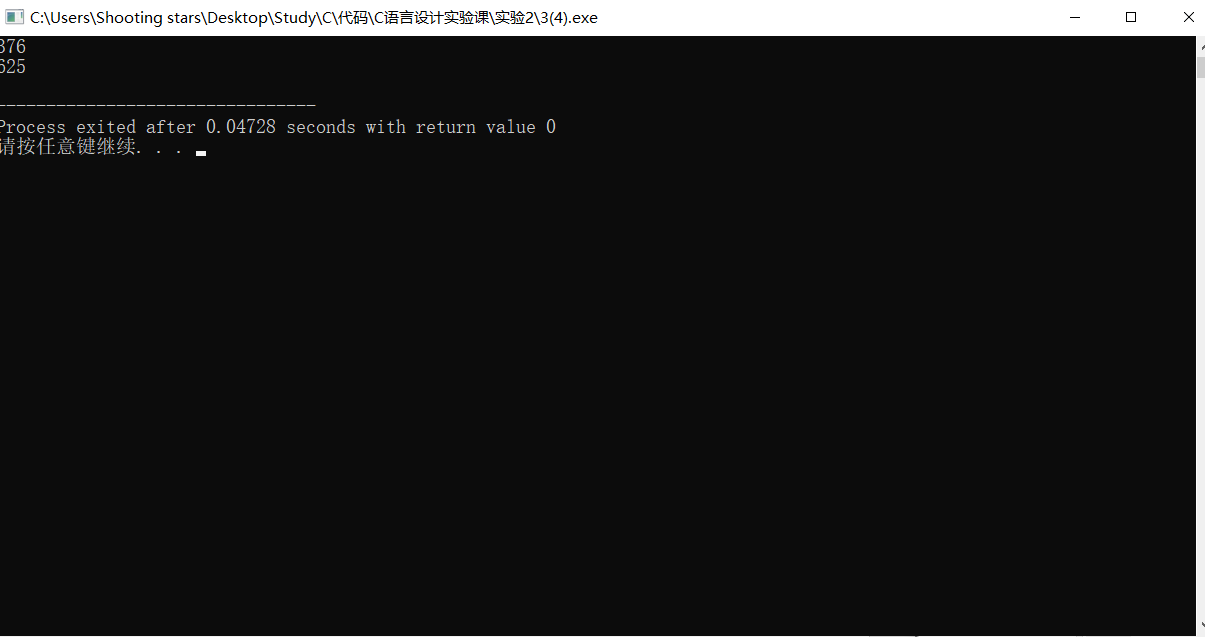


图2-2-9 运行结果示意图

2.3. 选做题

（1）判断给定的字符串是否是合法的C整型常量，如果是，则输出Yes；不是，则输出No。例如，0xabL是C整形常量，而092不是C整形常量。要求程序能够循环接受用户的输入，每行输入一个字符串，给出判定结果，直至输入Ctrl+Z结束。

解答：

代码：

#include <stdio.h>

int is(int);

int is8(int);

int is16(int);

char s[100];

int main()

{

int flag,i,fu;

while((scanf("%s", &s)) != EOF)

{

flag =1;

fu = 0;

i = 0;

if (s [0] == '-')

{

i++;

fu = 1;

}

if( s[fu] == '0')

{

i++;

if(s[fu+1] =='x'|| s[fu+1] == 'X')

{

i++;

flag = is16(i);

}

else flag = is8(i);

}

else

flag = is10(i);

if(flag)

printf("YES!");

else printf("NO!");

}

return 0;

}

int is10(int i)

{

int flag = 1;

while( s[i] !='\0')

{

if(s[i] >='0' && s[i]<= '9')

i ++;

else if(s[i] == 'u'|| s[i] == 'U')

{

i++;

if(s[i] == 'l'|| s[i] == 'L')

{

i++;

if(s[i] == '\0')

break;

else flag = 0;

}

else if(s[i] == '\0')

break;

else flag = 0;

}

else if(s[i] == 'l'|| s[i] == 'L')

{

i++;

if(s[i] == '\0')

break;

else flag = 0;

}

else if(s[i] == '\0')

break;

else

{

flag = 0;

break;

}

}

return flag;

}

int is8(int i)

{

int flag = 1;

while( s[i] !='\0')

{

if(s[i] >='0' && s[i]<= '7')

i ++;

else if(s[i] == 'u'|| s[i] == 'U')

{

i++;

if(s[i] == 'l'|| s[i] == 'L')

{

i++;

if(s[i] == '\0')

break;

else flag = 0;

}

else if(s[i] == '\0')

break;

else flag = 0;

}

else if(s[i] == 'l'|| s[i] == 'L')

{

i++;

if(s[i] == '\0')

break;

else flag = 0;

}

else if(s[i] == '\0')

break;

else

{

flag = 0;

break;

}

}

return flag;

}

int is16(int i)

{

int flag = 1;

while( s[i] !='\0')

{

if((s[i] >='0' && s[i]<= '9') ||(s[i] >='A' && s[i]<='F') ||(s[i]>='a' && s[i]<='f'))

i ++;

else if(s[i] == 'u'|| s[i] == 'U')

{

i++;

if(s[i] == 'l'|| s[i] == 'L')

{

i++;

if(s[i] == '\0')

break;

else flag =0;

}

else if(s[i] == '\0')

break;

else flag = 0;

}

else if(s[i] == 'l'|| s[i] == 'L')

{

i++;

if(s[i] == '\0')

break;

else flag =0;

}

else if(s[i] == '\0')

break;

else

{

flag = 0;

break;

}

}

return flag;

}

运行示例：

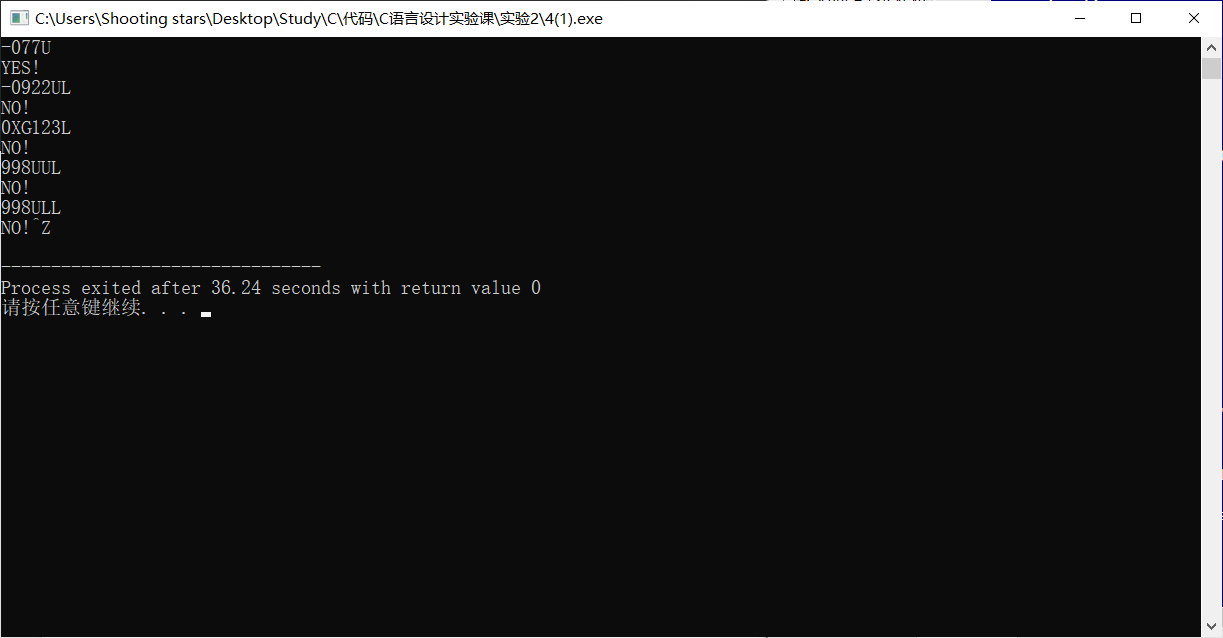


图2-3-1运行结果示意图

（2）输入正整数x（2⩽x⩽79），输出所有形如abcde/fghij=x的表达式，其中a～j由不同的数字0～9组成。例如：x=32时，输出为：75168/02349=32。

解答：

代码：

#include <stdio.h>

int main()

{

int n,x,y,flag,i,j,x1,y1;

scanf("%d", &n);

for(x = 12345;x<=98765;x++)

{

int a[10] = {0};

i=0;

flag = 1;

x1 =x;

if( x % n ==0)

{

y = x/n;

y1 = y;

for(i = 0;i<=4;i++)

{

a[i] = x1%10;

x1 /=10;

}

for(i=5;i<=9;i++)

{

a[i] = y1%10;

y1 /=10;

}

for(i = 0;i<=9;i ++)

{

for(j = 0;j <=9;j++)

{

if(i != j)

{

if(a[i] == a[j])

{

flag = 0;

break;

}

}

}

if(flag == 0)

break;

}

if (flag == 1)

printf("%05d/%05d=%d\n", x,y,n);

}

}

return 0;

}

运行示例：

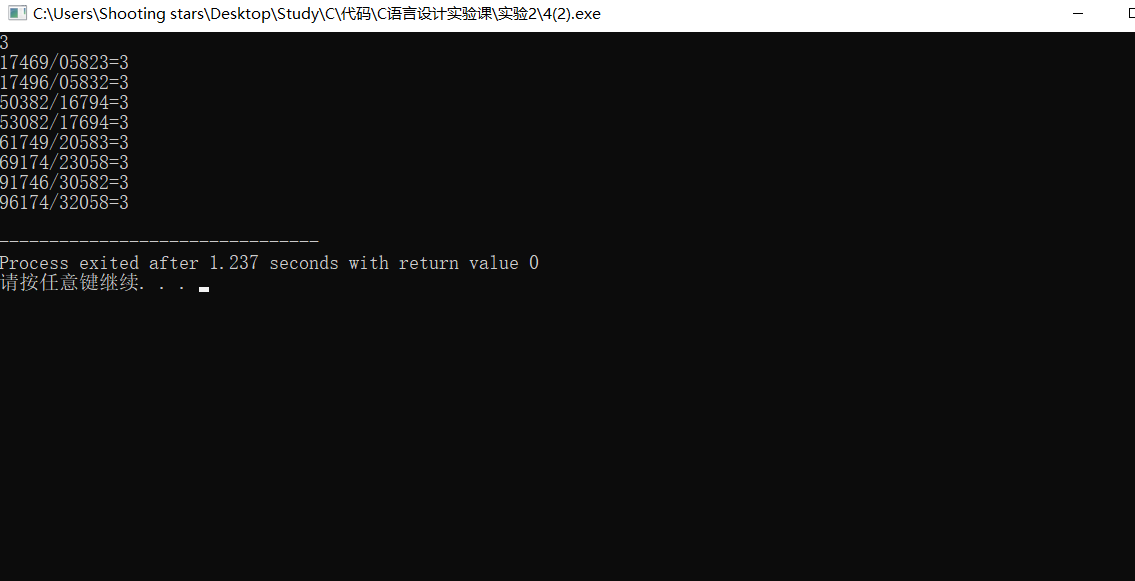


图2-3-2 运行结果示意图

**2.4.自设题**

题目描述：国王将金币作为工资，发放给忠诚的骑士。第一天，骑士收到一枚金币；之后两天（第二天和第三天），每天收到两枚金币；之后三天（第四、五、六天），每天收到三枚金币；之后四天（第七、八、九、十天），每天收到四枚金币……；这种工资发放模式会一直这样延续下去：当连续N天每天收到N枚金币后，骑士会在之后的连续N+1*N*+1天里，每天收到N+1*N*+1枚金币。

请计算在前K*K*天里，骑士一共获得了多少金币。

输入格式：一个正整数K*K*，表示发放金币的天数。

输出格式：一个正整数K*K*，表示发放金币的天数。

解答

框图：

****

图2-4-1程序设计流程图

代码：

#include<stdio.h>

int main()

{

int k,n,i,j,s,t;

scanf("%d", &k);

n = 0,s = 0, i =1;

while( s<k)

{

for(j = 1; j<=i; j++)

{

n += i;

s +=1;

if (s == k)

break;

}

i++;

}

printf("%d", n);

return 0;

}

运行示例：

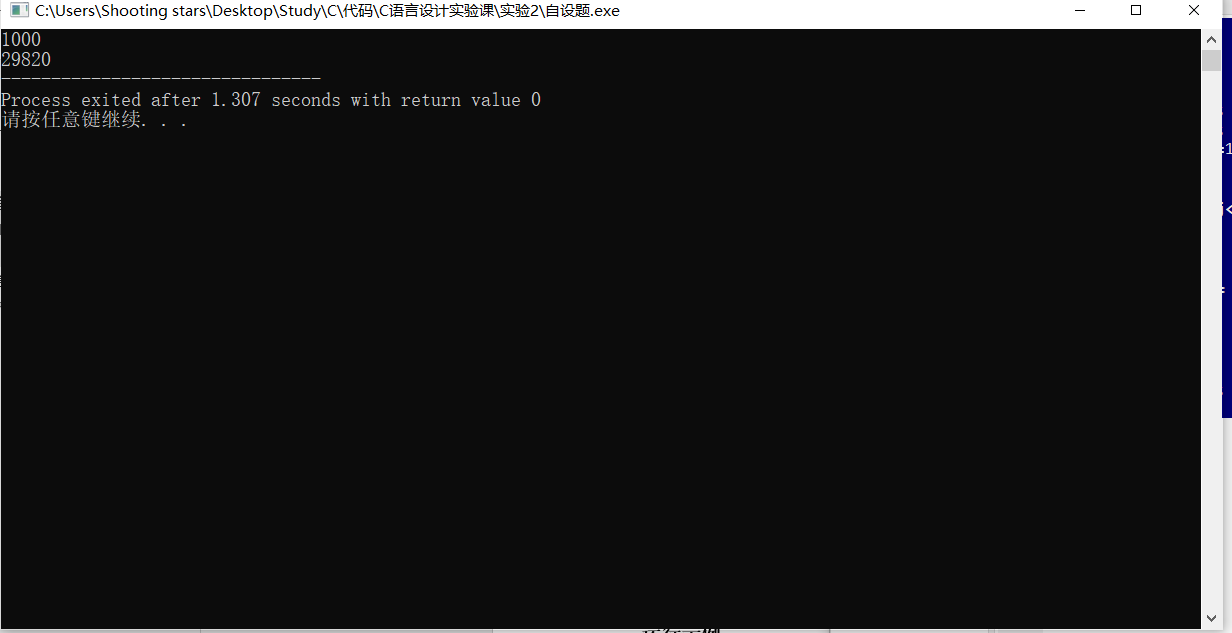


图2-4-2 运行结果示意图

## 2.5 实验小结

主要叙述实验过程中遇到的问题，如何解决的，通过分析、结果问题后的体会。

在实验过程中，有一些算法比较复杂，有许多重复的步骤，可以通过合并，break等语句进行省略，极大地提高了运算效率，使代码的可读性更强，更有利于处理复杂的问题。

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