根据需求编写了一下代码：

#include <stdio.h>

int main()

{

int num,a,b,c;

int temp;

for (num = 100;num < 300; num++)

{

temp = num;

//一边求最低位，一边右移一位。

**temp = (temp - (a = num % 10))/10;**

**temp = (temp - (b = temp % 10))/10;**

**c = temp % 10;**

//Debug

//printf("a = %d;b = %d;c= %d.\n",a,b,c);

//回来求个位数与百位数对调的数

temp = a \* 100 + b \* 10 + c;

if (temp > 2 \* num && temp < 3 \* num) printf("%d\n", num);

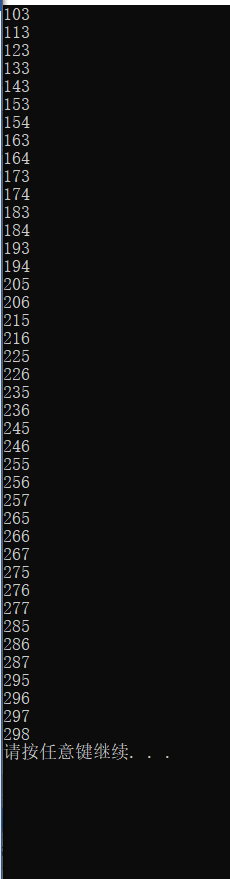
}

system("Pause");

return 0;

}

得到的结果很长：



随意选几个数字检验，结果正确。但是这样的排版不便于阅读。因此检查数据处改为：

if (temp > 2 \* num && temp < 3 \* num)

{

printf("%4d", num);

count++;

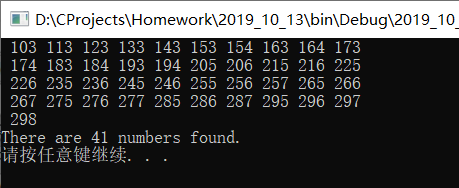
if(count%10 == 0) printf("\n");

}

（需要在起始处添加count定义）并在末尾加上

printf("\nThere are %d numbers found.\n",count);

这样得到的结果就是：



结果正确且清晰明了。

不妨再加点料：让用户决定枚举范围。

int min,max;

printf("Please enter two number to set searching range.\n");

scanf("%d %d",&min,&max);

当然，用当前所学知识可以进行有效性判断：

while(min<100 || min>999 || max<100 || max>999)

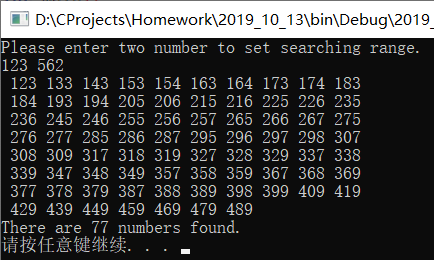
{

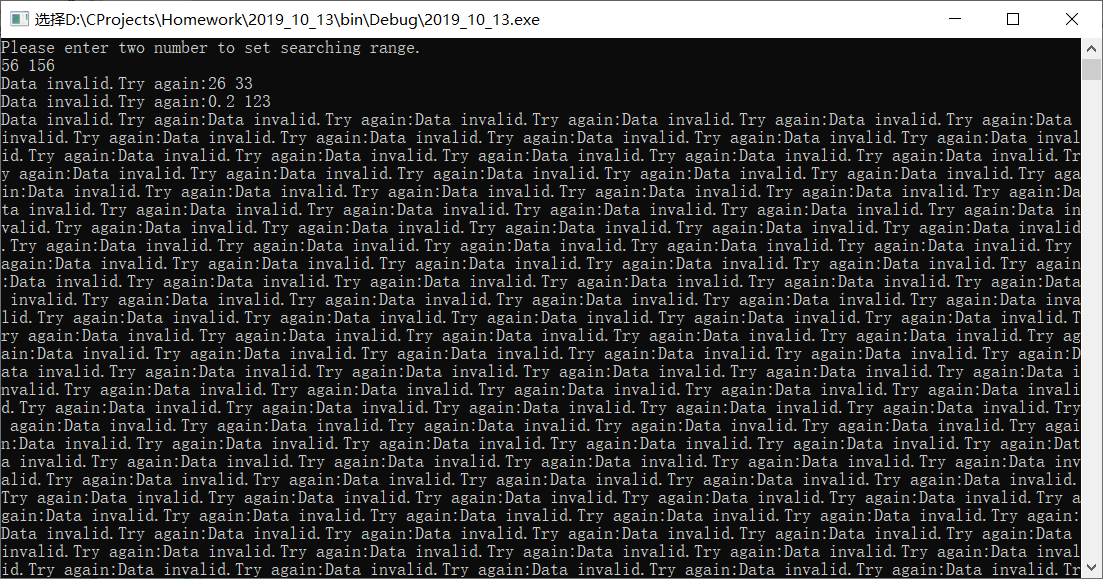
printf("Data invalid.Try again:");

scanf("%d %d",&min,&max);

}

正确数据测试：

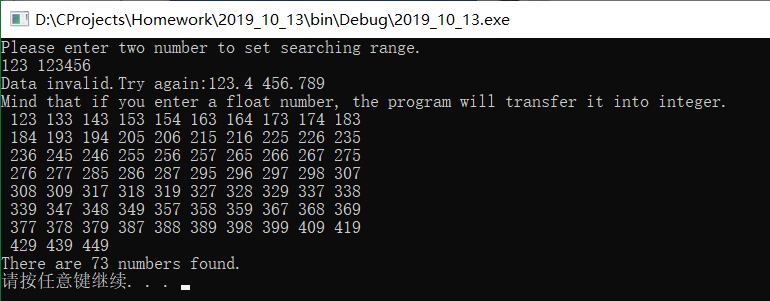


错误数据测试1：

输入错误会重复要求输入，但是输入小数会出现不明错误。猜测与Scanf有关。

解决方案：将min max改为float，并在恰当位置进行类型转换。

错误数据测试2：



正确地接住了浮点数，并进行相应处理。

最终代码如下：

#include <stdio.h>

int main()

{

int num,a,b,c,temp,count = 0;

float min = 100,max = 300;

int i\_min,i\_max;

printf("Please enter two number to set searching range.\n");

scanf("%f %f",&min,&max);

while(min<100 || min>999 || max<100 || max>999)

{

printf("Data invalid.Try again:");

scanf("%f %f",&min,&max);

}

//直接赋值

i\_min = min;i\_max = max;

//提示可能会有类型转换

printf("Mind that if you enter a float number, the program will transfer it into integer.\n");

if(i\_min>i\_max)

{

//用位异或操作排个序

i\_min = i\_min ^ i\_max;

i\_max = i\_min ^ i\_max;

i\_min = i\_min ^ i\_max;

}

for (num = i\_min;num < i\_max; num++)

{

temp = num;

//一边求最低位，一边右移一位。

temp = (temp - (a = num % 10))/10;

temp = (temp - (b = temp % 10))/10;

c = temp % 10;

//Debug

//printf("a = %d;b = %d;c= %d.\n",a,b,c);

//回来求个位数与百位数对调的数

temp = a \* 100 + b \* 10 + c;

if (temp > 2 \* num && temp < 3 \* num)

{

printf("%4d", num);

count++;

if(count%10 == 0) printf("\n");

}

}

printf("\nThere are %d numbers found.\n",count);

system("Pause");

return 0;

}

--2019-10-13