西安交通大学实验报告

课程名称： 算法设计与问题求解 实验名称： 字符串

学 院： 机械工程学院 实验日期 2020 年 10 月 15 日

班 级： 机械97班 姓 名： 杨逢诜 学号： 2193712613

#### 一、实验内容和结果

* **题目1 字符串处理**

考虑char 数组表示的字符串，比如str[]=” Abc 789 ”，现在要去掉该字符串头部、尾部的空格，中间的空格保留。

【源程序】

#include<stdio.h>

#include<string.h>

void TrimLeft(char\* string) //声明函数：空格删除

{

int i; //声明计数变量，用于循环计数

for(i=strlen(string)-1;;i--) //从字符串最后一个字符开始，删去全部空格，直至碰到一个非空格字符；

{

if(string[i]==' ')

{

string[i]=0;

}

else break;

}

int k=0; //声明计数变量：用于计算前部空格数量

for(i=0;;i++) //从第一个字符开始，计量开头空格数目，直至碰到一个非空格字符；

{

if(string[i]==' ')

{

k++;

}

else break;

}

for(i=0;;i++) //将空格后的字符串内容整体前移，平移空间与空格数相同，在字符串终止符处停止；

{

if(string[i+k-1]==0)

{

break;

}

string[i]=string[i+k];

}

}

int main() //主函数开始；

{

char string[80]; //声明并输入字符串；

printf("Please input the string:");

gets(string);

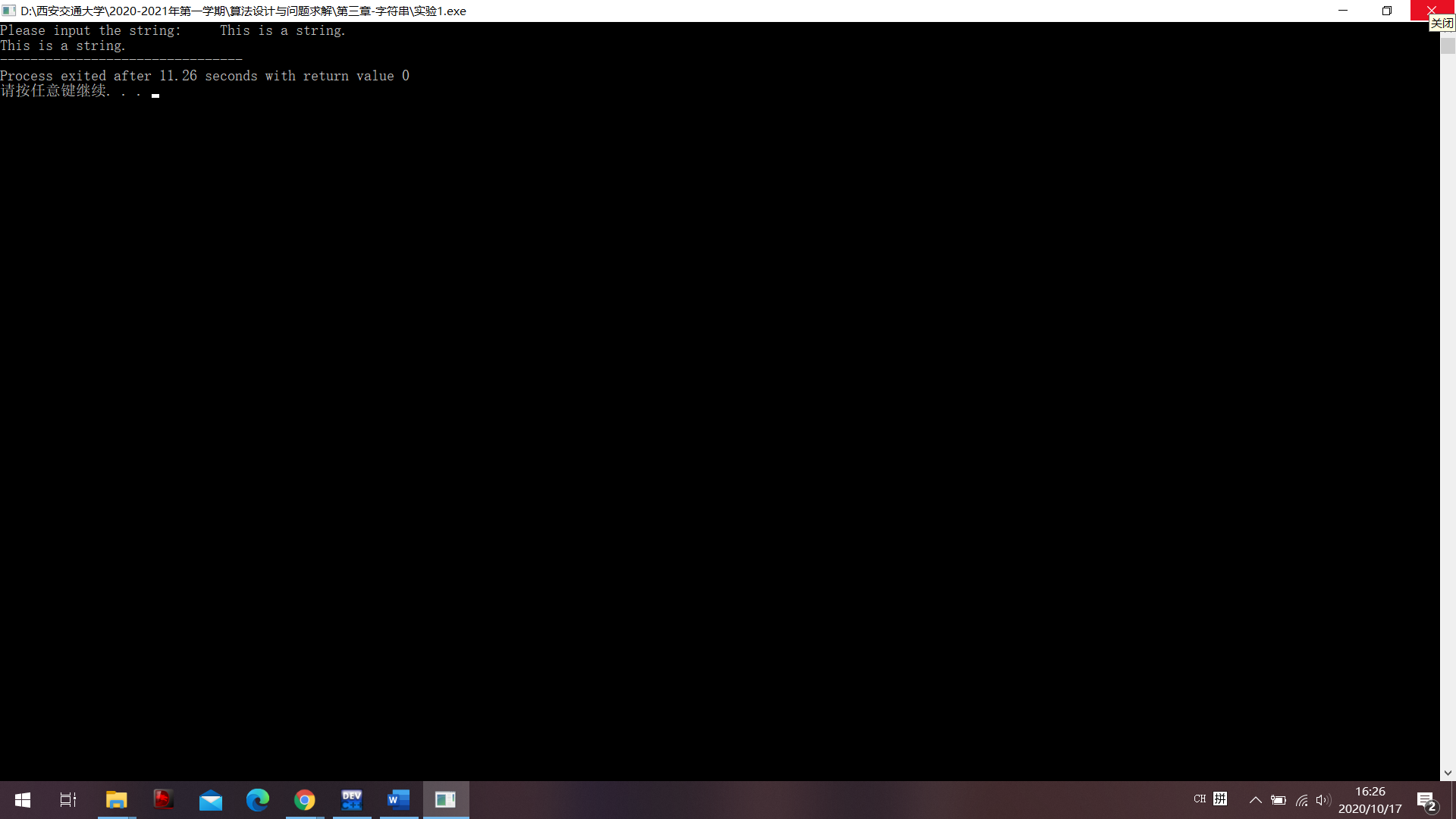
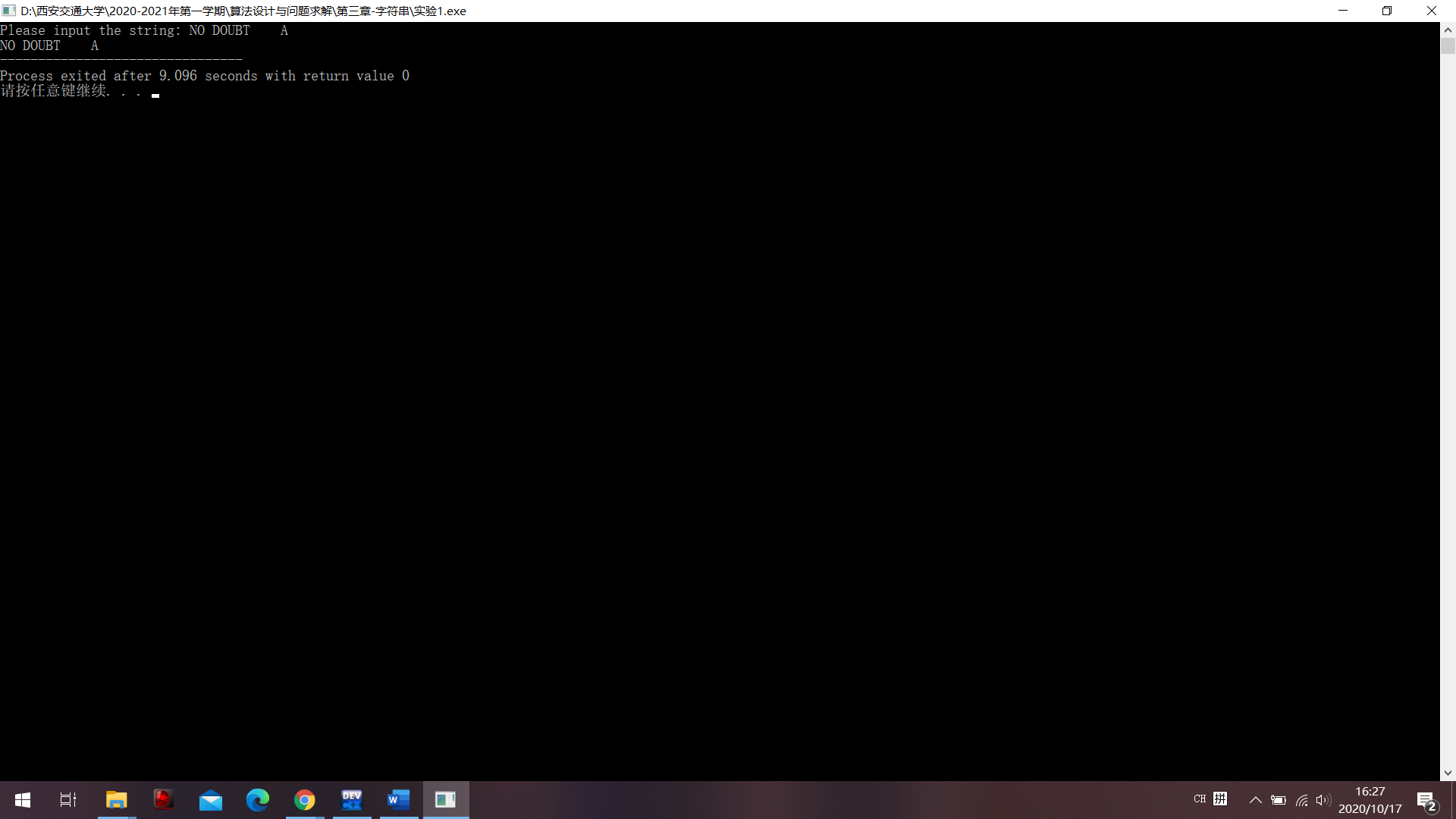
TrimLeft(string); //删去额外空格，输出对应字符串，主函数结束

printf("%s",string);

return 0;

}

【运行结果】

* **题目2 将字符型数字变为真实数字**

考虑char 数组表示的正整数数字，比如”123”、”976”，现在要将他们变为int型数字，然后将两个数字作加法。

【源程序】

#include<stdio.h>

#include<string.h>

int ConvertToInt(char\* string) //声明函数：数字的字符串-整数转换

{

int k=0,i; //声明变量：k作为最终计算得到的整数，i作为辅助计数变量用于控制循环；

for(i=0;;i++)

{

if(string[i]==0) //终止条件：读取到字符串终止符

{

break;

}

k=k\*10+string[i]-'0'; //施行进位和补位操作：\*10使得已输入的数字进位，+string[i]-'0'将下一位数字输入到进位后的最后一位中

}

return k; //返回转换得到的整数；

}

int main() //主函数开始

{

char str[20]; //声明数组、整数：前一个用于存放待输入的字符串，后一个用于存放转换得到的整数

int num;

printf("Please input the Integer:"); //按照提示输入数字字符串

gets(str);

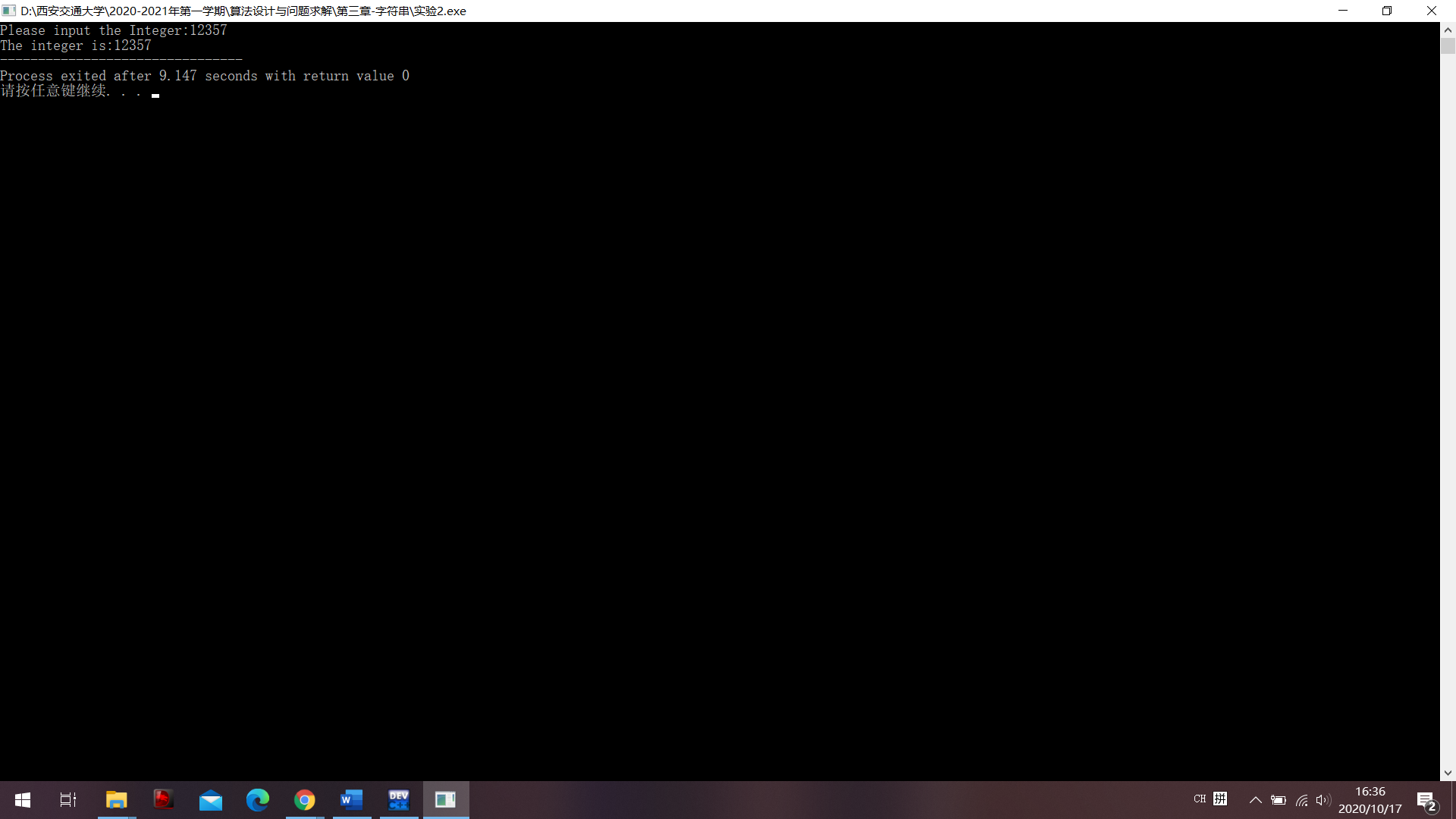
num=ConvertToInt(str); //对字符串实行转换，输出转换结果，主函数结束

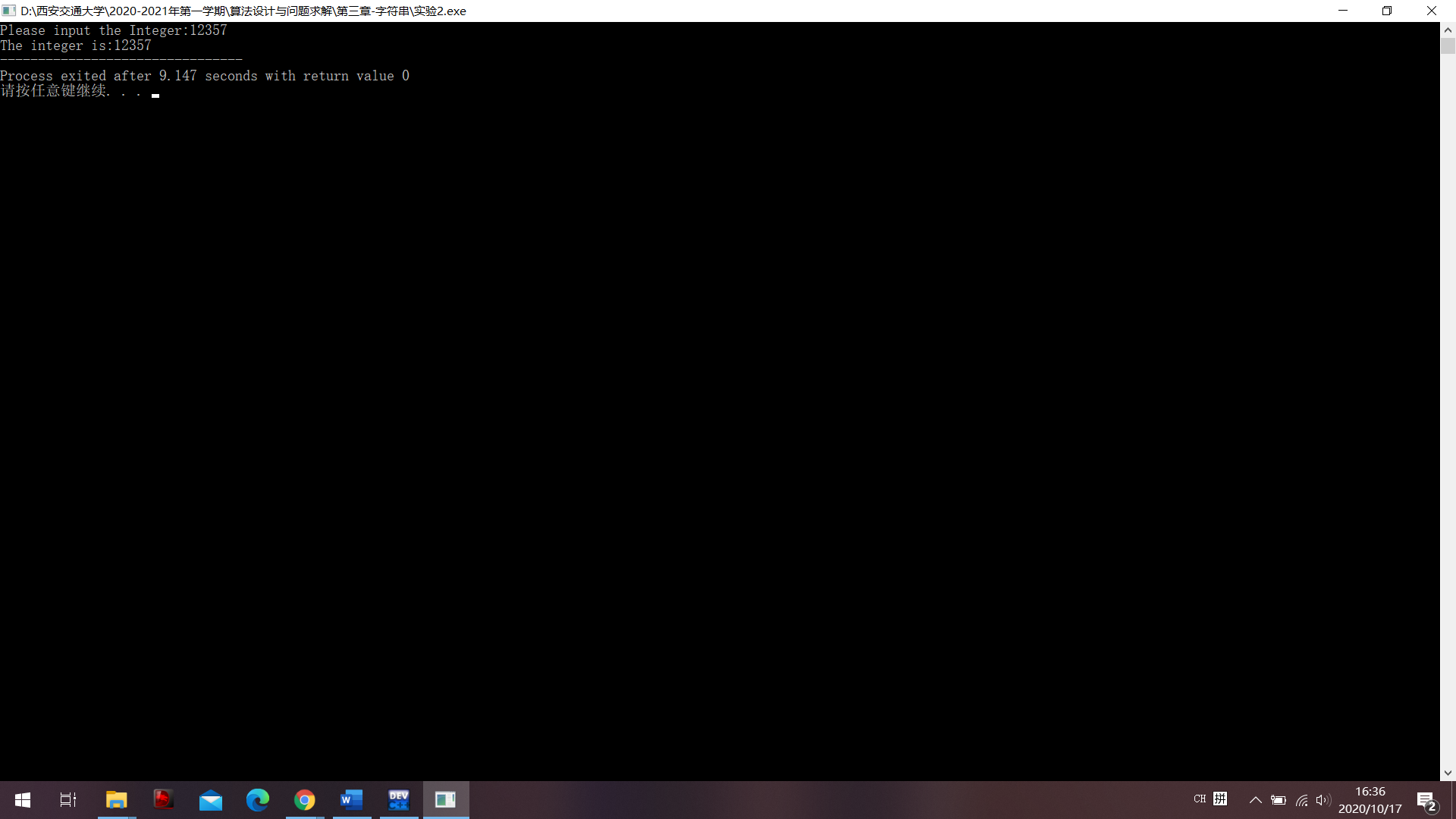
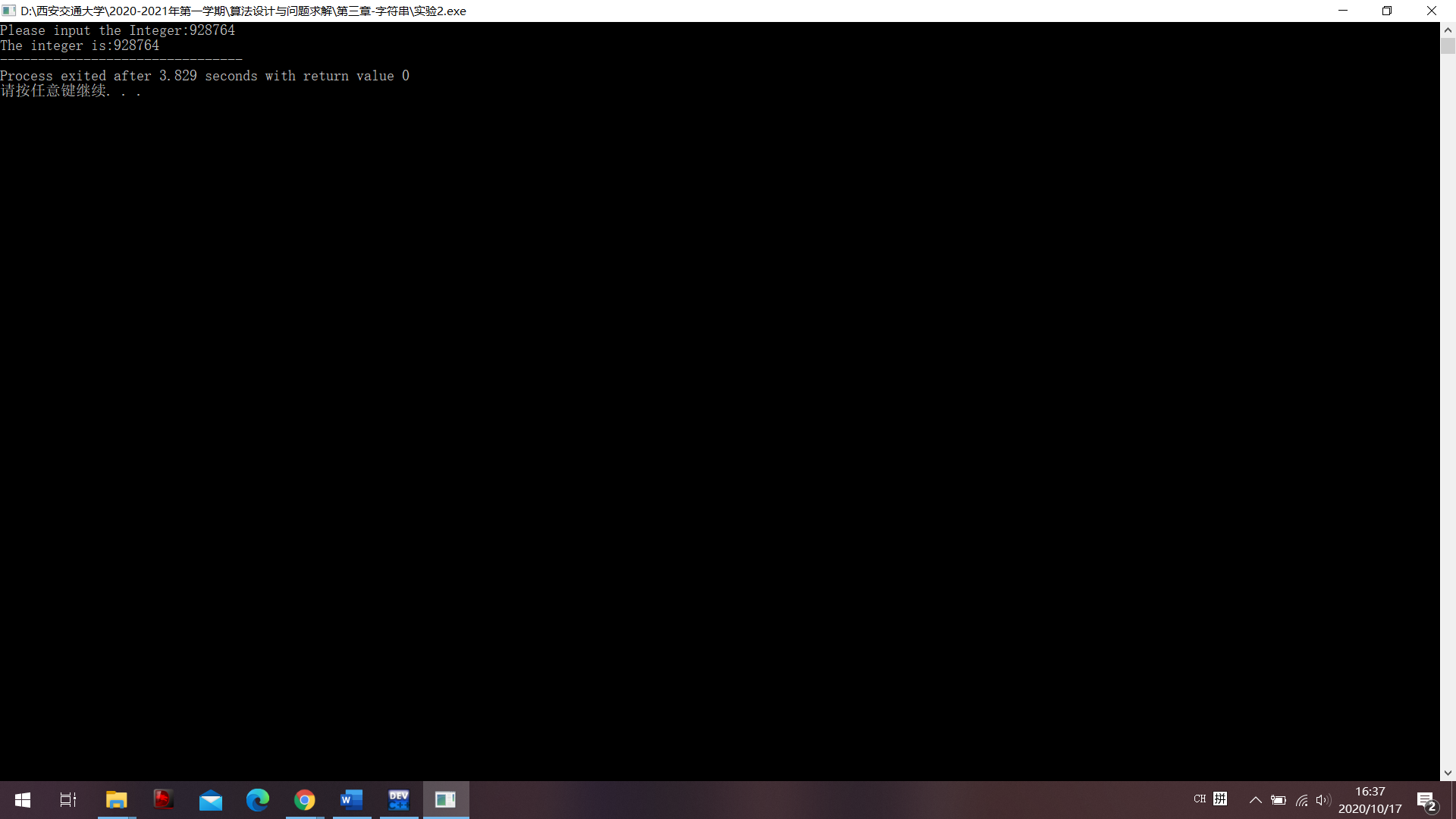
printf("The integer is:%d",num);

return 0;

}

【运行结果】



* **题目3 判断正整数是否左右对称**

所谓对称数，就是譬如：1，323，34543之类的数。可以把原来的数的高低位互换，构造一个新的数，如果和原来的数相等则是对称数。

【源程序】

#include<stdio.h>

#include<string.h>

int main() //主函数开始

{

char num[20],numrev[20]; //声明数组，以字符串数组的形式存储数字；

int i; //声明计数变量，用于控制循环；

printf("Please input the integer:"); //输入数字；

gets(num);

numrev[strlen(num)]=0; //构造反转数，首先在数组的合适位置添加终止符；

for(i=0;i<strlen(num);i++) //使得原数第一位成为反转数的最后一位，原数第二位成为反转数的倒数第二位，以此类推；

{

numrev[i]=num[strlen(num)-i-1];

}

if(strcmp(num,numrev)==0) //比较反转前后两字符串是否存在差异；

{

printf("This integer is reversable."); //若反转前后两字符串相同，则称该数字是对称的，否则称之为不对称的；

}

else

{

printf("This integer is irreversable.");

}

return 0; //主函数结束

}

【运行结果】

