计算机与信息工程学院实验报告

••••••••••••••••••••••••••••••••• 密 ••••••••••••••••••••••••••••••••• 封 ••••••••••••••••••••••••••••••••• 线 •••••••••••••••••••••••••••••••••

姓名：\_\_扶靖棋\_\_\_ 学号：\_\_\_2012020028\_\_\_\_\_

专业：\_\_计算机科学与技术\_\_ 年级：\_\_\_2020级\_\_\_

课程： 汇编语言与接口技术 主讲教师：\_\_王玉璟\_\_

辅导教师：\_\_\_王玉璟

实验时间：\_2020\_\_年 \_11\_月 \_2\_\_日 \_下\_午\_17\_时至\_19\_时，实验地点\_\_\_608\_\_\_\_\_

实验题目： 实验二 分支循环结构程序练习

实验目的： 1. 通过编制包含分支、循环结构的程序，练习汇编语言综合编程的方法。2.练习在Debug状态下调试程序的方法

实验环境（硬件和软件） i5-1035G1 CPU 16.0 GB Windows10

软件：DOXBOX

实验内容：

1、试编写程序，完成以下功能。

（1）程序执行时，显示提示信息“Please input a string(length<9):”，由用户输入一个长度小于9的字符串；

（2）然后显示提示信息“Please input the index of the char to display:”，请用户指定该串中的某个字符的位置号，程序控制用户输入的位置号必须是合法的，例如实际输入字符串长度为5个字符，位置号只能是0-4之间的数值；若位置号不合法，则程序退出。

（3）程序将用户指定位置的字符显示出来。

**程序输出样例1：**

Please input a string(length<9): ABCDEFG

Please input the index of the char to display: 2

The char is: C

**程序输出样例2：**

Please input a string(length<9): ABCD

Please input the index of the char to display: 4

The index is invalid！

2、试使用分支结构和循环结构程序完善上面的程序功能。

（1）采用01号DOS功能调用，接收用户输入的字符串，并在程序中对字符串进行长度检查，若长度大于9，则要求用户重新输入；

（2）用户在指定位置号时，进行判断，若输入非法（位置号大于实际字符串长度），则提示用户重新输入；若输入位置号合法，则显示字符串中该位置号对应的字符。

（3）统计用户输入的字符串中特定字符的数目并显示。可先指定特定字符进行统计，然后再由用户指定字符进行统计。

**程序输出样例1：**

Please input a string(length<9): ABCD

Please input the index of the char to display: 1

The char is: B

Please input a char: A

The count of ‘A’ is : 1

**程序输出样例2：**

Please input a string(length<9): ABCDEFGHIJ

The string is too long!

Please input a string(length<9):ABCDEF

Please input the index of the char to display: 9

The index is invalid！

Please input the index of the char to display: 3

The char is: D

Please input a char: 1

The count of ‘1’ is : 0

实验步骤：

编程思路（思想）：

使用系统功能调用输出指定的字符串以及获取用户输入的字符串以及想要查询的位置号。对位置号进行判断，若它大于等于输入字符串的长度，则跳转到事先编写好一段指令使程序输出“The index is invalid!”之后程序结束，否则继续向下执行，获取对应位置的字符然后进行输出，然后无条件跳转至程序结束。

源码

       DATA SEGMENT

    STR\_1 DB 0DH, 0AH,'Please input a string(length<9): $'

    STR\_2 DB 0DH, 0AH,'Please input the index of the char to display: $'

    STR\_3 DB 0DH, 0AH,'The char is: $'

    STR\_4 DB 0DH, 0AH,'The index is invalid!$'

    STR\_5 DB 0DH, 0AH,'Please input a char: $'

    STR\_6 DB 0DH, 0AH,'The string is too long! $'

    STR\_7 DB 0DH, 0AH,'The CNT\_1 of ',27H, ?, 27H, ' is:', 0DH, 0AH, '$'

    STR\_8 DB 0DH, 0AH,'MyInformation:2012020028\_fujingqi$'

    BUFF DB 20 DUP(?)

    CNT DB ?

    LEN DW ?

DATA ENDS

STACK SEGMENT

    DB 2,5,6

STACK ENDS

CODE SEGMENT

    ASSUME CS:CODE, DS:DATA, ES:DATA, SS:STACK

START:  MOV AX, DATA

        MOV DS, AX

        MOV ES, AX

        MOV AX, STACK

        MOV SS, AX

        MOV SP, 3

INPUT:  LEA DX, STR\_1   ;输出提示输入字符串

        MOV AH, 9

        INT 21H

        XOR BX,BX       ;异或清零

        LEA SI, BUFF

                        ;输入一个字符

GET:    MOV AH,01       ;存入AL

        INT 21H

        MOV [SI], AL

        INC SI

        INC BX

        CMP AL, 0DH

        JNZ GET         ;若AL与0D不相等，则转移到GET

                        ;若AL不为回车则跳转，继续输入字符

        CMP BX, 9

        JLE INDEX       ;BX小于等于9则转移INDEX

        LEA DX, STR\_6

        MOV AH, 9       ;输出字符串，提示输入字符串长度太长

        INT 21H

        JMP INPUT       ;无条件跳转到输入口，再次输入

INDEX:  MOV LEN, BX

        DEC LEN         ;LEN存储字符串长度

        LEA DX, STR\_2

        MOV AH, 9       ;输出字符串，提示输入索引下标

        INT 21H

        MOV AH, 01

        INT 21H

        MOV CL, AL      ;输入索引

        MOV DX, LEN     ;取实际长度

        ADD DL, 30H

        CMP CL, DL

        JGE ERROR       ;索引大于等于实际长度，跳转输出错误

        CMP CL, 30H     ;索引小于0，跳转输出错误

        JL ERROR

        LEA DX, STR\_3

        MOV AH, 9       ;输出字符

        INT 21H

        SUB CL, 30H

        XOR CH, CH

        MOV SI, CX

        LEA BX, BUFF

        MOV DL, BYTE PTR[BX + SI - 1]

                        ;输出指定下标字符

        MOV AH, 2

        INT 21H

        LEA DX, STR\_5

        MOV AH, 9       ;输出字符串

        INT 21H

        MOV AH, 01

        INT 21H         ;输入待统计字符

                        ;将待统计字符存入DATA

        LEA BX, CNT

        MOV BYTE PTR[BX], AL

        LEA BX, STR\_7

        MOV BYTE PTR[BX+16], AL

        LEA DX, STR\_7

        MOV AH, 9       ;输出字符串

        INT 21H

                        ; 设置循环次数

        MOV CX, LEN

        XOR CH, CH

        XOR DL, DL

        LEA SI, CNT

        MOV AL, BYTE PTR[SI]

                        ; AL中存待统计字母,CL存循环次数，DL中存数量

        LEA BX, BUFF

NEXT:   CMP BYTE PTR[BX], AL

        JE CNT\_1

IDX:    INC BX

        LOOP NEXT

        ADD DL, 30H

        MOV AH, 2

        INT 21H

                        ;用DL统计数

        MOV DL, 30H

        JMP RETURN  ;程序结束

ERROR:  LEA DX, STR\_4

        MOV AH, 9 ;输出字符串

        INT 21H

        JMP INDEX

CNT\_1:  INC DL

        JMP IDX

RETURN: LEA DX, STR\_8

        MOV AH, 9       ;输出字符串，提示输入索引下标

        INT 21H

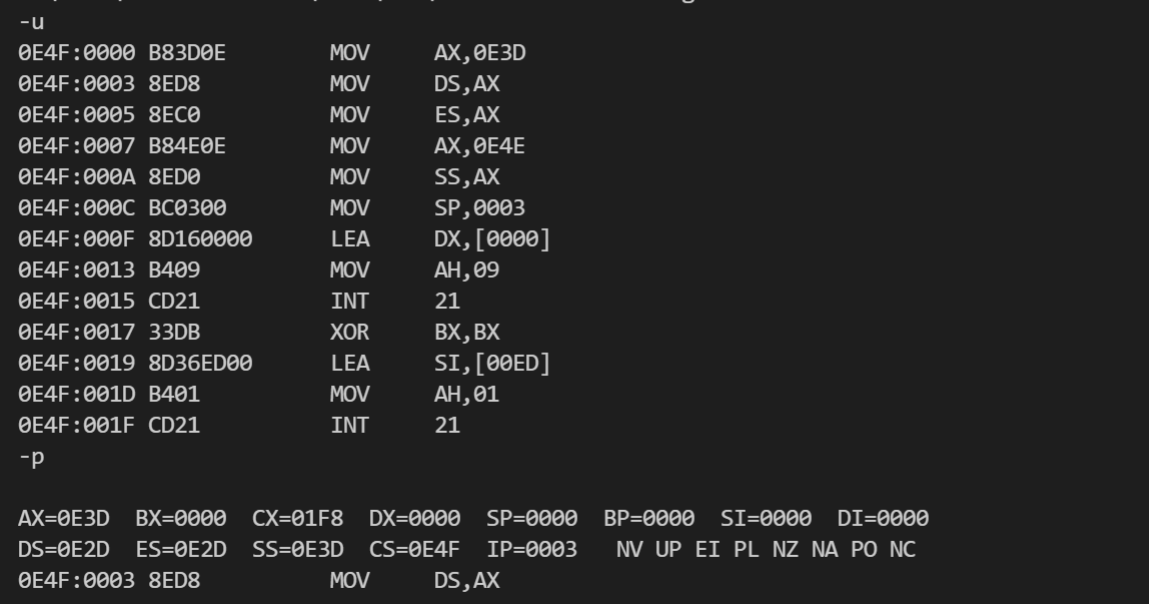
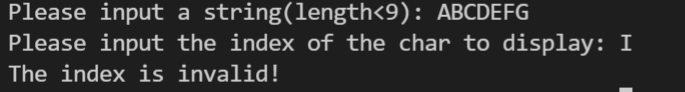
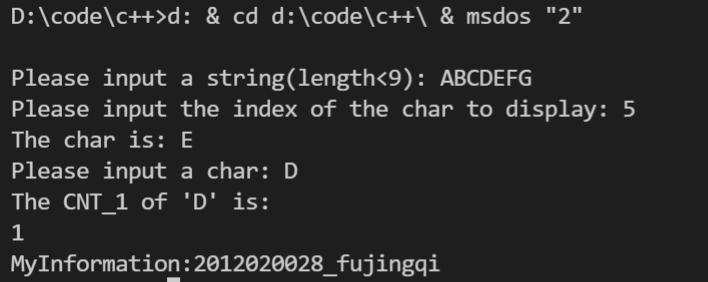
        MOV AH, 4CH ;

        INT 21H

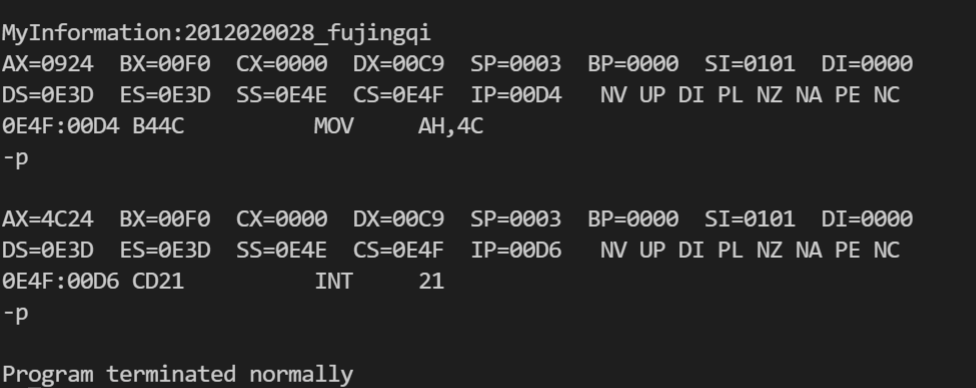
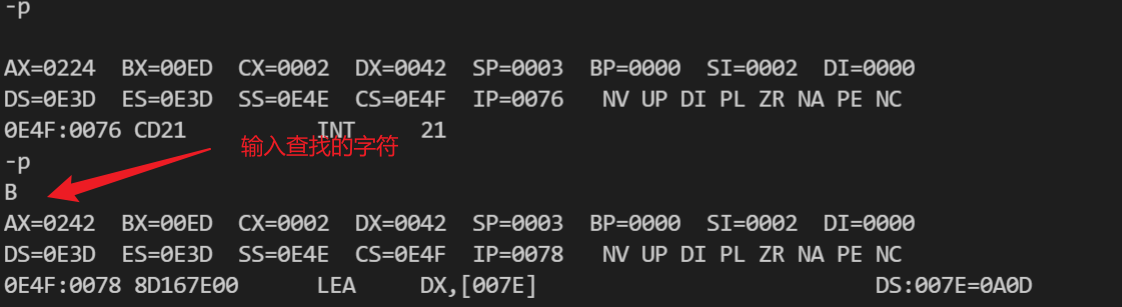
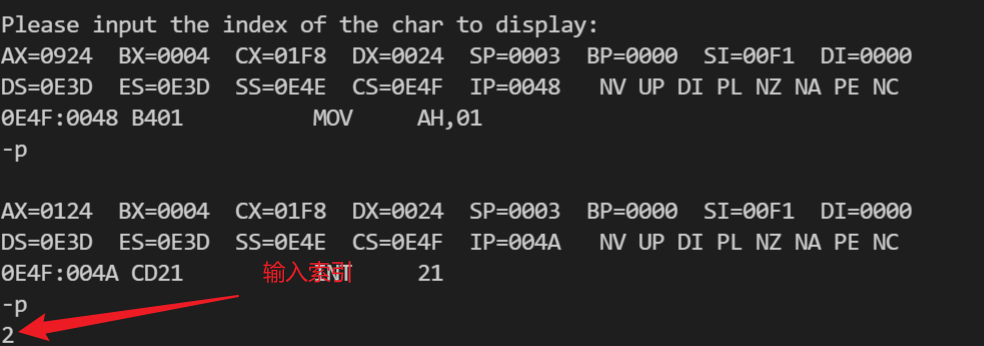
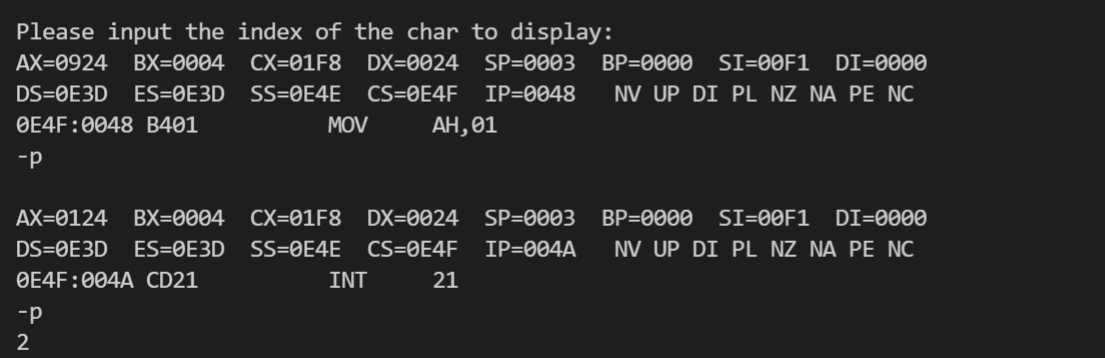
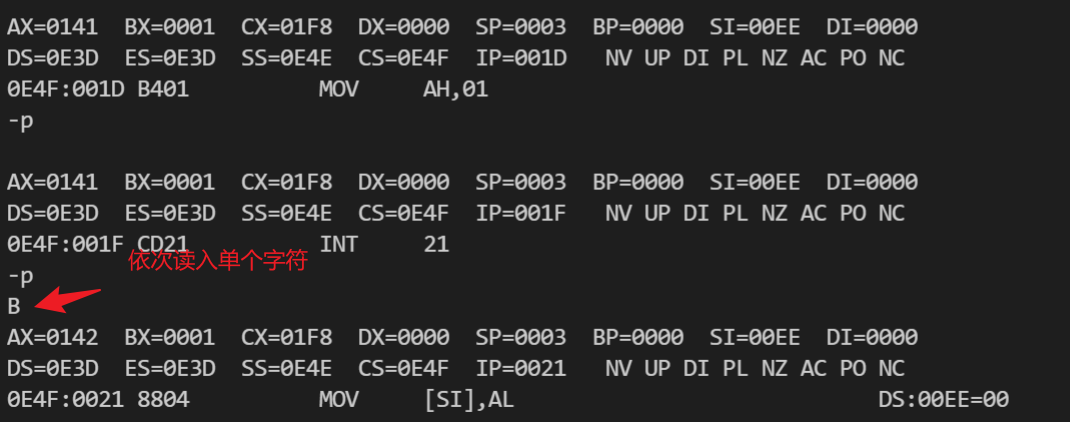
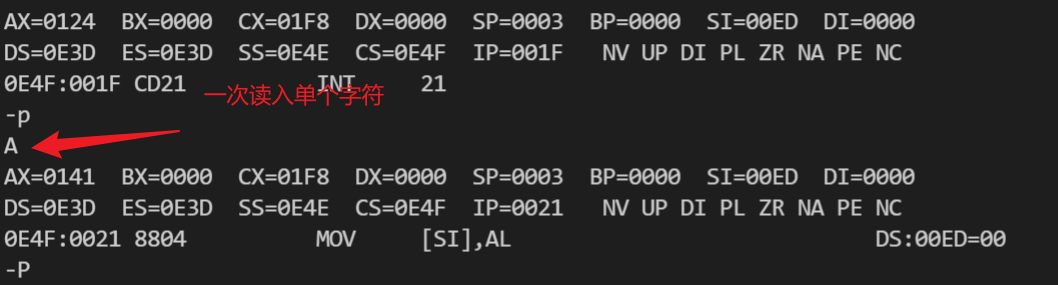
CODE ENDS

END START

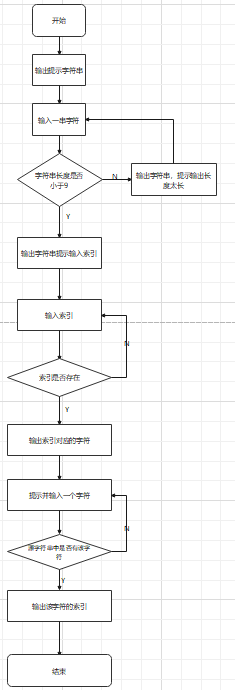
实验数据记录：



·



流程图：



问题讨论：

**问题1：**如何用01号功能实现字符串的输入

**答：**用跳转语句和读入实现，只要不是回车就重复读入，不断执行这个操作直到最后读取到回车为止。

**问题2**：如何将输入的字符插入提示字符中

**答：**事先在对应的提示字符中开辟一个未被使用的存储空间，之后在得到用户输入的数据之后，将这个数据插入到对应的地址即可。

**问题3**：找到字符串指定的偏移地址位置时报错

**答：**通过查阅相关书籍和网站可以知道：AX、CX、DX不能用于间接寻址，BX，SI等这些特定的基址和变址寄存器才可用于间接寻址。DX跟AX这些16位寄存器只可用于存放地址，数据以及用作系统的一些特定功能。

**问题4**：调用dos的各种功能后，dx中的数值发生了改变，导致本来寄存在其中的有效数据丢失，从而造成程序结果的错误。

**答：**dos功能调用既然会导致dx中值发生改变，所以只要先开辟适当大小的内存空间将所需要的数据存储进去就可以避免数据的丢失。

**问题5：**如何遍历每个字符，并统计所要找的字符

**答：**在读入时候，我们其实已经获取了这个字符串的长度，那么之后只要把这个长度对应到循环次数上，并把计数器也设置为偏移量，之后就可以从字符串的最后一个字符一直向前，直到遍历完所有字符串中的字符，如果出现了和用户输入相同的字符，令我们提前设计的计数器加一即可。

**问题6：**如何判断字符串的长度是否不符合规范？

**答**：使用cmp指令对获取的字符串长度是否在有效范围内做出判断，如若字符串长度在有效范围内，则继续往下执行，否则提示输入的字符串过长并返回输入。