****

**汇编语言程序设计课程设计**

****

|  |  |
| --- | --- |
| **题目** | **子程序与输入输出程序设计** |
| **学院** | **计算机学院** |
| **专业** | **计算机科学与技术** |
| **学号** |  |
| **姓名** |  |
| **指导老师** | **余波老师** |
| **实验时间** | **2022年5月31日** |

# 第五次实验：子程序与输入输出程序设计

学号: 姓名:

## 一、实验目的：

掌握子程序设计方法，常用系统调用的使用方法，能够设计字符串处理相关的程序。

## 二、实验环境（硬件，软件环境）：

硬件环境： MACOS系统

软件环境：DOSBox v0.74-3-3

## 三、实验内容及要求（算法，程序，步骤，方法）：

**1、实验内容**

（1）原理分析：子程序调用、返回原理；系统调用相关原理

（2）完整的程序流程图、源代码

（3）完整的测试结果（测试所用字符串在后面给出）

**2、实验要求**

设计一个完整的源程序，实现以下功能：

（1）从键盘输入（带回显）一个长度在10字符以内的字符串，并保存在缓冲区；

（2）使用子程序实现：检测并删除字符串中的数字字符，将字符串中所有小写字母改为大写字母；

（3）将处理后的字符串在新的一行显示，若字符串为空串则显示 ‘？’

**3、实验步骤**

**（1）原理分析**

要实现输入字符串，将该字符串删除数字并转换小写字母为大写字母的子功能，可以分析出该源程序需要三个系统调用指令——输入字符串，输出一个字符（输出字符为‘?’的情况），输出字符串（输出转换结果）以及两个子程序——删除数字，转换小写字母为大写字母。

本题中在同一个代码段进行书写，为防止写错，故本题中使用子程序调用指令CALL

本题中使用系统调用指令02H，0AH，09H

以下给出相应子程序调用返回以及系统调用的相关功能并指出原理。

**3.1 子程序调用、返回原理**

**3.1.1 段内直接调用**

1.格式：CALL OPR

2.执行操作：先保存断点：S P ←S P-2，将CALL的下一条指令的IP入栈；在将子程序名OPR代表的偏移地址→IP，转到子程序执行。

3.功能：子程序名直接写在指令中，作段内调用。

**3.1.2 段内间接调用**

1.格式：CALL WORD PTR OPR

2.执行操作：将断点处的IP入栈保存；如果子程序的偏移地址在16位寄存器中则把寄存器的内容→IP；如果其偏移地址是用存储器中的一个字指出，则把改存储器单元的内容→IP

3.功能：子程序的偏移地址由寄存器或存储单元指出，作段内调用。

**3.1.3 段间直接远调用**

1.格式：CALL FAR PTR OPR

2.执行操作：先将CALL的下一条指令的CS和IP分别入栈；再把子程序的偏移地址→IP，子程序所在段的段地址→CS

3.功能：子程序名用FAR PTR属性直接写在指令中，作跨段调用。

**3.1.4 段间间接调用**

1.格式：CALL DWORD PTR OPR

2.执行操作：先将CALL的下一条指令的CS和IP分别入栈；再把存储单元的(EA)→IP，（EA+2）→CS

3.功能：子程序名保存在双字单元中，第一个字作偏移地址，第二个字作为段地址，做跨段调用。

**3.1.5 返回指令RET**

1.格式：RET [n]

2.执行操作：

（1）段内返回（近返回）时，从堆栈段中弹出的断点仅修改IP；

（2）段间返回（远返回）时，从堆栈段中弹出断点的偏移地址→IP，再弹出断点的段地址→CS；

（3）如果是RET n指令，表示弹出断点后，再将堆栈段指针SP+n之后再返回。

3.功能：用于子程序中，返回到主程序的断点处继续执行。执行时，将断点从堆栈中弹出，修改IP或修改IP、CS。

**3.2 系统调用相关原理**

系统调用拥有自己的系统调用号，系统调用表以及系统调用处理函数。一个调用号对应一个处理函数，通过处理函数和调用号在调用表中找到相应的系统调用。

这里主要使用了INT 21H即DOS提供的服务子程序，下面列举实验要求中所使用的系统调用号。

**3.2.1 21H号调用的02H号子功能——显示单个字符**

1.功能：将DL中指定的字符在当前光标位置显示，光标向右移动一个字符位置，自动换行，滚屏。

2.实验源码中用法：用以显示单个空字符‘？’的情况

**3.2.2 21H号调用的0AH号子功能——输入字符串并回显**

1.功能：从键盘接收多个字符，将字符按照接收顺序由低地址到高地址保存在指定的缓冲区中。

2.实验源码中用法：用以接收输入的字符串

**3.2.3 21H号调用的09H号子功能——显示字符串**

1.功能：显示指定缓冲区中内的字符串，光标移动到字符串末尾，显示直到遇到‘$’字符。

2.实验源码中用法：用以输出转换后的字符串

**3.2.4 21H号调用的4CH号子功能——中断指令**

1. **完整的程序流程图、源代码**

由于程序需要两个子程序功能，故分开画：

**3.2.1 程序流程图**

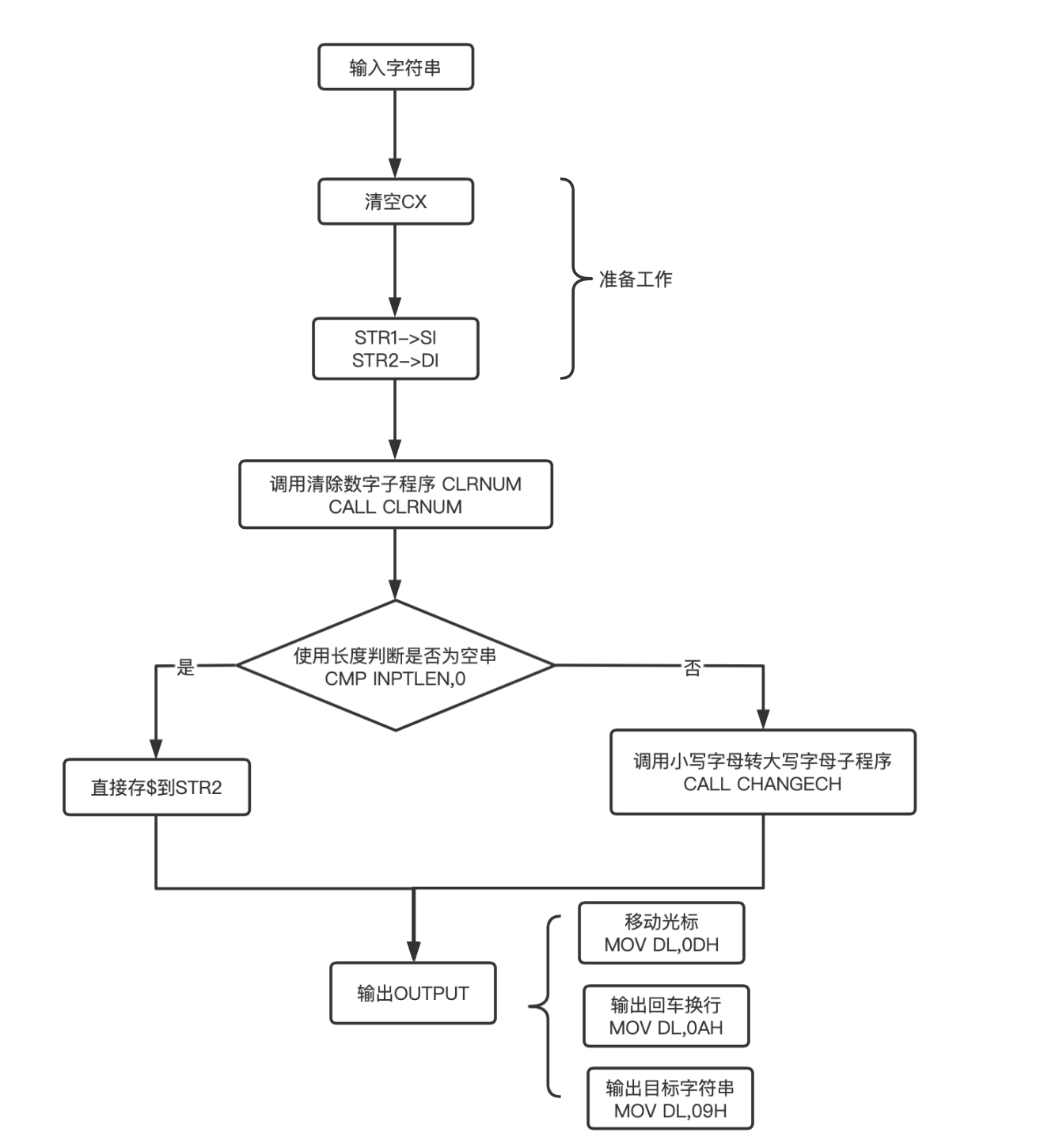


图1 主程序流程图

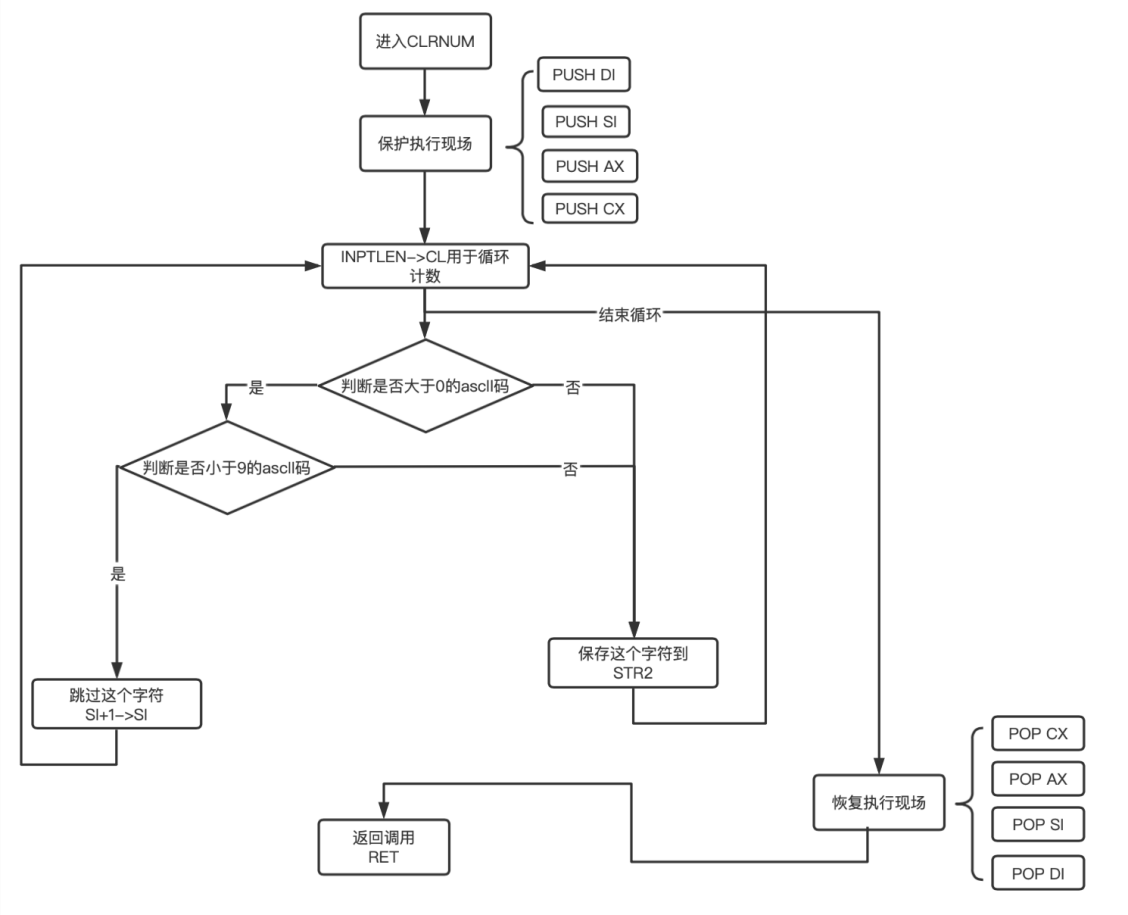


图2 CLRNUM流程图

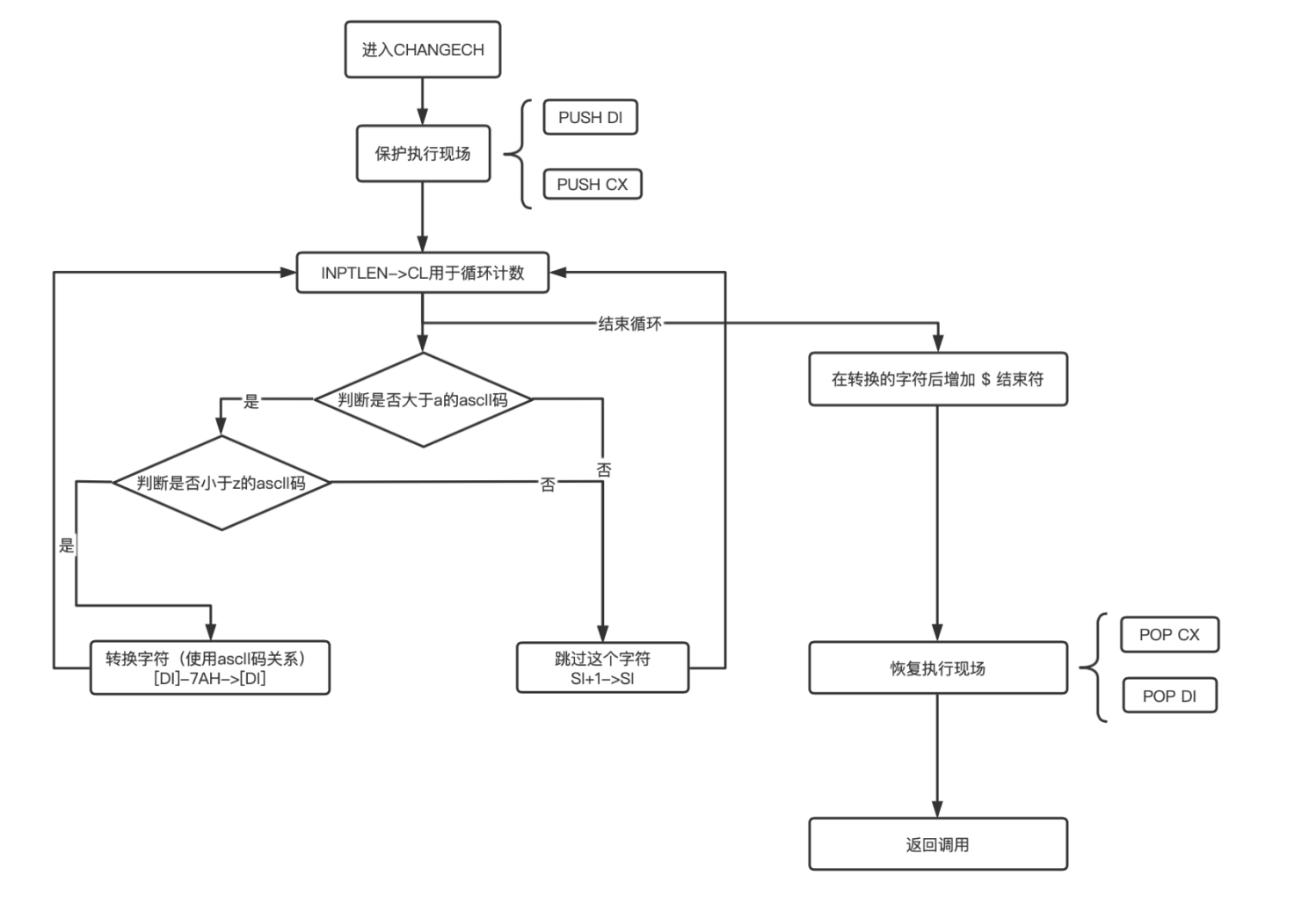


图3 CHANGECH流程图

**3.2.2 程序源代码**

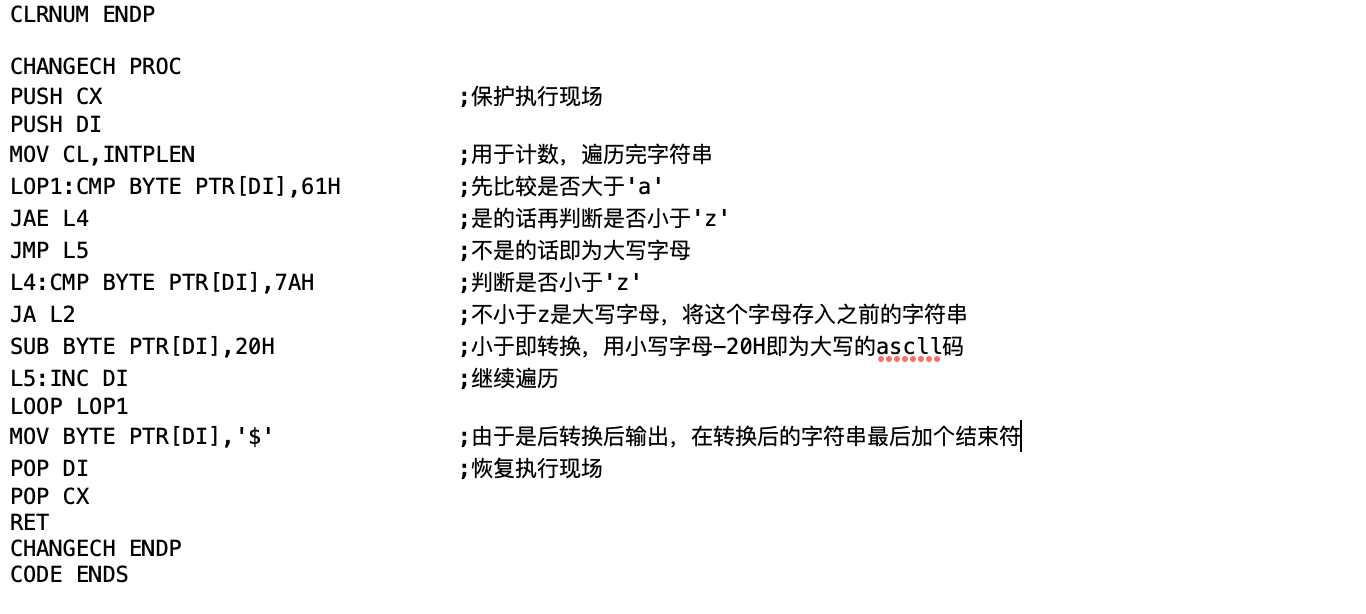
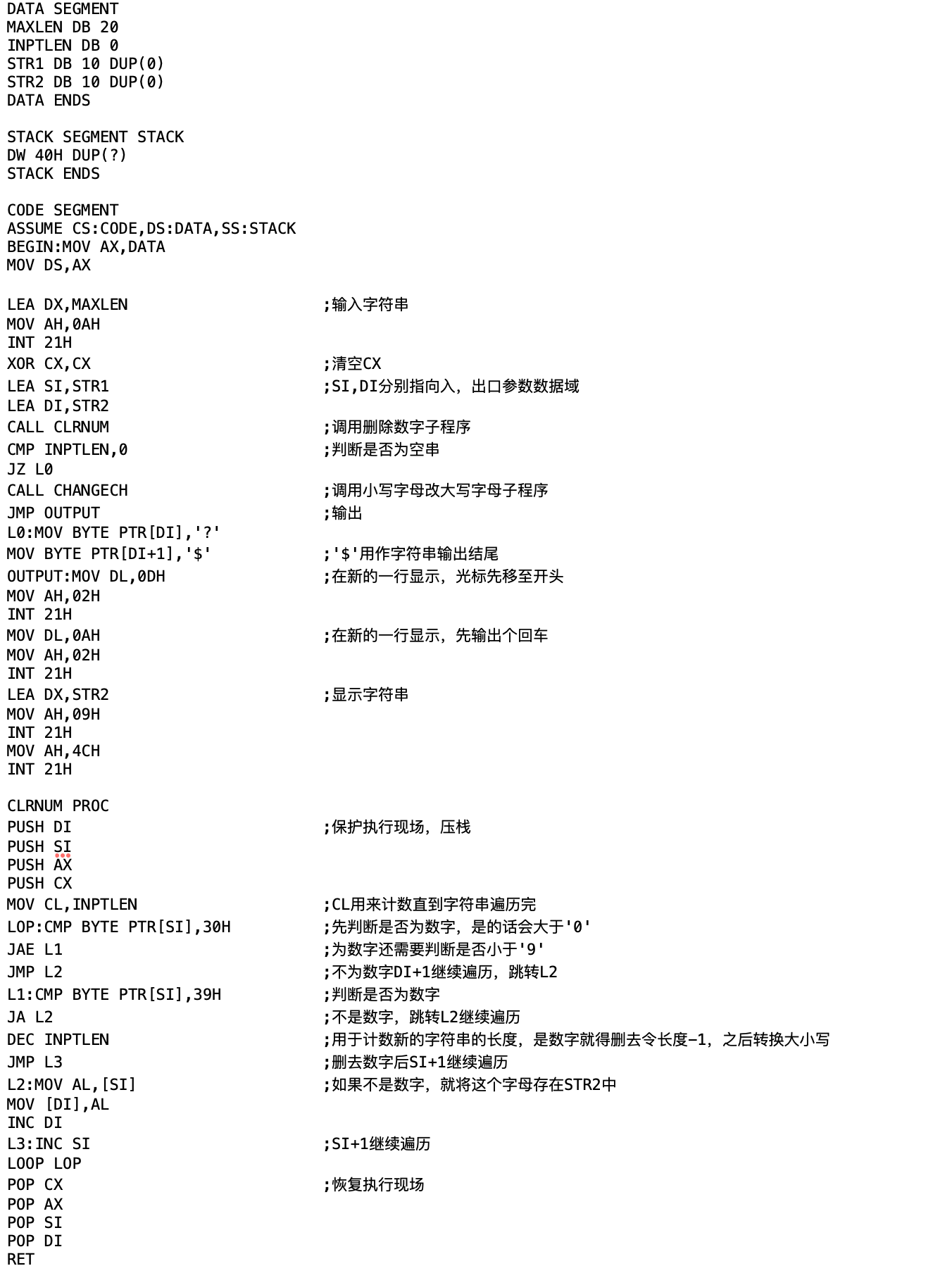


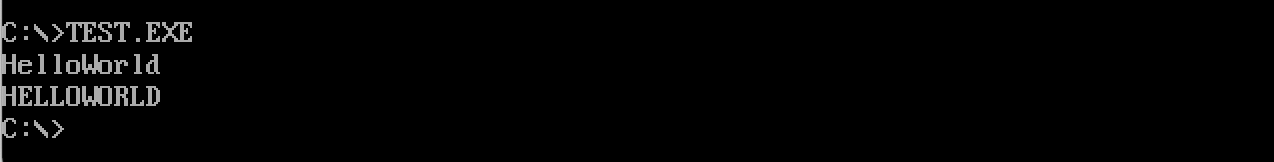
图4 源代码（含注释）

1. **完整的测试结果**

按照实验要求后直接编译（masm），链接（link）后运行源程序（直接输入源程序名称TEST.EXE即可）

测试字符串分别为：HelloWorld，Mike 1989，9999

测试结果如下：



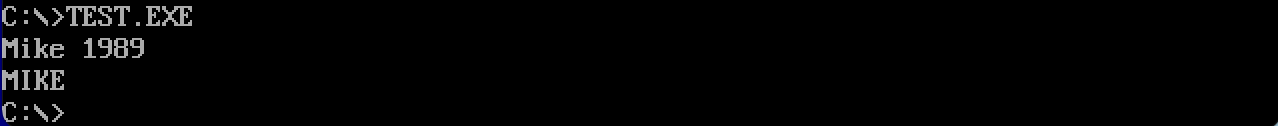




图5 测试结果

## 4、实验结果分析：

（1）对于HelloWorld，其中不含数字，只转换小写字母为大写字母，最终转换结果应为HELLOWORLD，实验验证结果符合实验预期。

（2）对于Mike 1989，其中含数字，删除数字后变为Mike ，之后转换小写字母为大写字母，最终转换结果应为MIKE ，实验验证结果符合实验预期。

（3）对于9999，全为数字，最后得到的应为空字符串输出“$”，实验验证结果符合实验预期。

综上，实验结果均符合实验预期，实验成功！

## 5、实验体会：

1.学会了如何编写子程序，一般子程序写在主程序之后

2.学会了如何输入输出字符串，掌握了简单的串处理指令

3.本题中子程序与主程序位于同一代码段，故要主要使用段内调用指令，但最好用CALL，可以自动匹配段内或者段间调用

4.记得返回调用RET

5.子程序记得保护执行现场，以免造成不必要的错误

6.注意输出时要加$符号作为结束符

7.程序较长时可以分块画流程图自上而下简化思路