

学号： WA2224013 专业： 机器人工程 姓名： 郭义月

实验日期： 2024. 12. 27 指导教师： 鲍华 实验成绩：

课程目标 1 (权重_____)	课程目标 2 (权重_____)	课程目标 3 (权重_____)	课程目标 4 (权重_____)	课程目标 5 (权重_____)	课程目标 6 (权重_____)	综合成绩 (目标数可增删)

安徽大学人工智能学院本科实验报告

【课程名称】 微型计算机原理及接口技术

【课程目标】

提升对内存操作、条件判断、循环控制与子程序的理解与应用能力。

【实验名称】 实验五 大小写转换与ASCII码输出

【实验目的】

利用汇编语言实现大小写转化与ASCII码输出的功能

【实验原理及方法】

在这两个汇编语言编程任务中，第一个任务涉及将字符串中的大写字母转换为小写字母并显示。程序通过遍历字符串，检查每个字符是否为大写字母（ASCII码值在65到90之间），如果是，则通过在其ASCII码值上加32（即20H）来转换为小写字母（ASCII码值在97到122之间），然后使用DOS的9号功能调用输出每个字符。第二个任务则是处理一组十六进制数，将它们转换成ASCII码并显示，每两个数换行。程序通过定义一个计数器来跟踪当前处理的十六进制数的位置，利用TEST指令和01H来检查计数器是否为偶数，如果是，则输出换行符以实现每两个数换行的效果。这两个任务展示了在汇编语言中如何进行字符处理和屏幕输出，包括字符的大小写转换和基于条件的格式化输出。

【实验内容及过程】

1. 设有字符STRING DB ‘I AM A STUDENT!’，编写程序将其中的大写字母转换成小写字母并用DOS 9号功能调用输出显示字符串。

```
DATAS SEGMENT
    ;此处输入数据段代码
    STR1 DB 'I AM A STUDENT!$'
DATAS ENDS

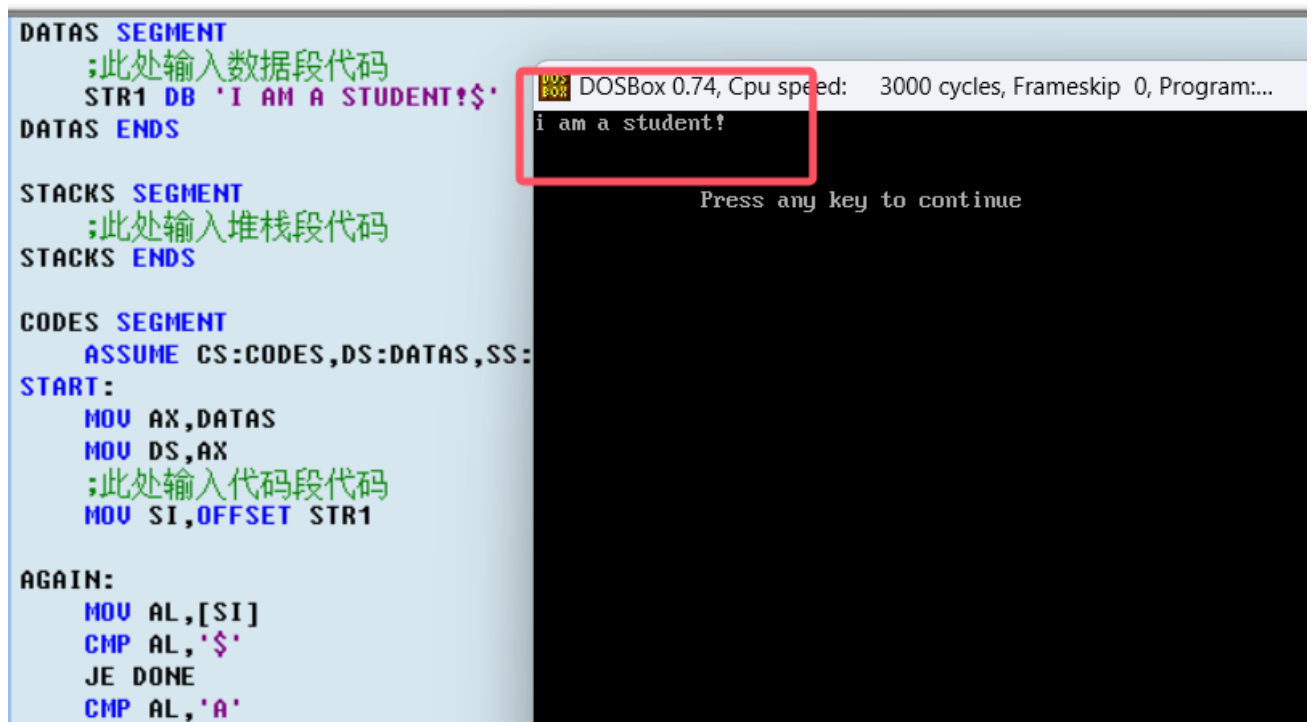
STACKS SEGMENT
    ;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
    ;此处输入代码段代码
    MOV SI,OFFSET STR1

AGAIN:
    MOV AL,[SI]
    CMP AL,'$'
    JE DONE
    CMP AL,'A'
    JB NEXT
    CMP AL,'Z'
    JA NEXT
    ADD AL,20H
    MOV [SI],AL
NEXT:
    INC SI
    JMP AGAIN
DONE:
    MOV DX,OFFSET STR1
    MOV AH,9
    INT 21H

    MOV AH,4CH
    INT 21H
CODES ENDS
END START
```

实验结果：



```

DATAS SEGMENT
;此处输入数据段代码
STR1 DB 'I AM A STUDENT!$'
DATAS ENDS

STACKS SEGMENT
;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
ASSUME CS:CODES,DS:DATAS,SS:
START:
MOV AX,DATAS
MOV DS,AX
;此处输入代码段代码
MOV SI,OFFSET STR1

AGAIN:
MOV AL,[SI]
CMP AL,'$'
JE DONE
CMP AL,'A'

```

2. 在内存中定义一组十六进制数，将其每位转换成ASCII码，然后显示，每两个数换一行。

```

DATAS SEGMENT
;此处输入数据段代码
DATA1 DB 41H,42H,43H,44H,45H,45H,47H,48H,49H,50H,58H
DATAS ENDS

STACKS SEGMENT
;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
MOV AX,DATAS
MOV DS,AX
;此处输入代码段代码

MOV CH,0
MOV SI,OFFSET DATA1
AGAIN:
MOV AL,[SI]
CMP AL,0
JE DONE
MOV DL,AL
MOV AH,02H
INT 21H
MOV DL,' '
MOV AH,02H

```

```

    INT 21H
    INC CH
    TEST CH,01H
    JNZ NEXT
    MOV DL,0AH
    MOV AH,02H
    INT 21H

NEXT:
    INC SI

    JMP AGAIN
DONE:
    MOV AH,4CH
    INT 21H
CODES ENDS
END START

```

实验结果:

The screenshot shows a DOSBox window with the following assembly code on the left:

```

DATAS SEGMENT
;此处输入数据段代码
DATA1 DB 41H,42H,43H,44H,45H,45H,47H,48H,49H,50H,58H
DATAS ENDS

STACKS SEGMENT
;此处输入堆栈段代码
STACKS ENDS

CODES SEGMENT
ASSUME CS:CODES,DS:DATAS
START:
MOV AX,DATAS

```

On the right, a character input menu is displayed, enclosed in a red box. The menu lists the following characters: A, B, C, D, E, E, G, H, I, P, X. The DOSBox status bar at the top right shows "DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro". At the bottom right, it says "Press any key to continue_".

【数据分析及处理】

实验1的思路为:

在循环里面依次遍历str1中的值, 存在si中, 判断si中的值是否在A-Z之间, 如果不在, 直接跳转到next中, 如果在A-Z之间, 就增加20H将其转化为小写字母

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...  
AX=0749 BX=0000 CX=003B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=0010 NU UP EI PL NZ NA PO NC  
0771:0010 720B JB 001A  
-p  
AX=0749 BX=0000 CX=003B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=0012 NU UP EI PL NZ NA PO NC  
0771:0012 3C5A CMP AL,5A  
-p  
AX=0749 BX=0000 CX=003B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=0014 NU UP EI NG NZ AC PO CY  
0771:0014 7704 JA 001A  
-p  
AX=0749 BX=0000 CX=003B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=0016 NU UP EI NG NZ AC PO CY  
0771:0016 0420 ADD AL,20  
-p  
AX=0769 BX=0000 CX=003B DX=0000 SP=0000 BP=0000 SI=0000 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=0018 NU UP EI PL NZ NA PE NC  
0771:0018 8804 MOV [SI],AL DS:0000=49  
-▲
```

I在A-Z之间，加20h从49变成69

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...  
AX=0720 BX=0000 CX=003B DX=0000 SP=0000 BP=0000 SI=0001 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=000A NU UP EI PL NZ NA PO NC  
0771:000A 3C24 CMP AL,24  
-p  
AX=0720 BX=0000 CX=003B DX=0000 SP=0000 BP=0000 SI=0001 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=000C NU UP EI NG NZ AC PE CY  
0771:000C 740F JZ 001D  
-p  
AX=0720 BX=0000 CX=003B DX=0000 SP=0000 BP=0000 SI=0001 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=000E NU UP EI NG NZ AC PE CY  
0771:000E 3C41 CMP AL,41  
-p  
AX=0720 BX=0000 CX=003B DX=0000 SP=0000 BP=0000 SI=0001 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=0010 NU UP EI NG NZ AC PO CY  
0771:0010 720B JB 001A  
-p  
AX=0720 BX=0000 CX=003B DX=0000 SP=0000 BP=0000 SI=0001 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=001A NU UP EI NG NZ AC PO CY  
0771:001A 46 INC SI  
-a_
```

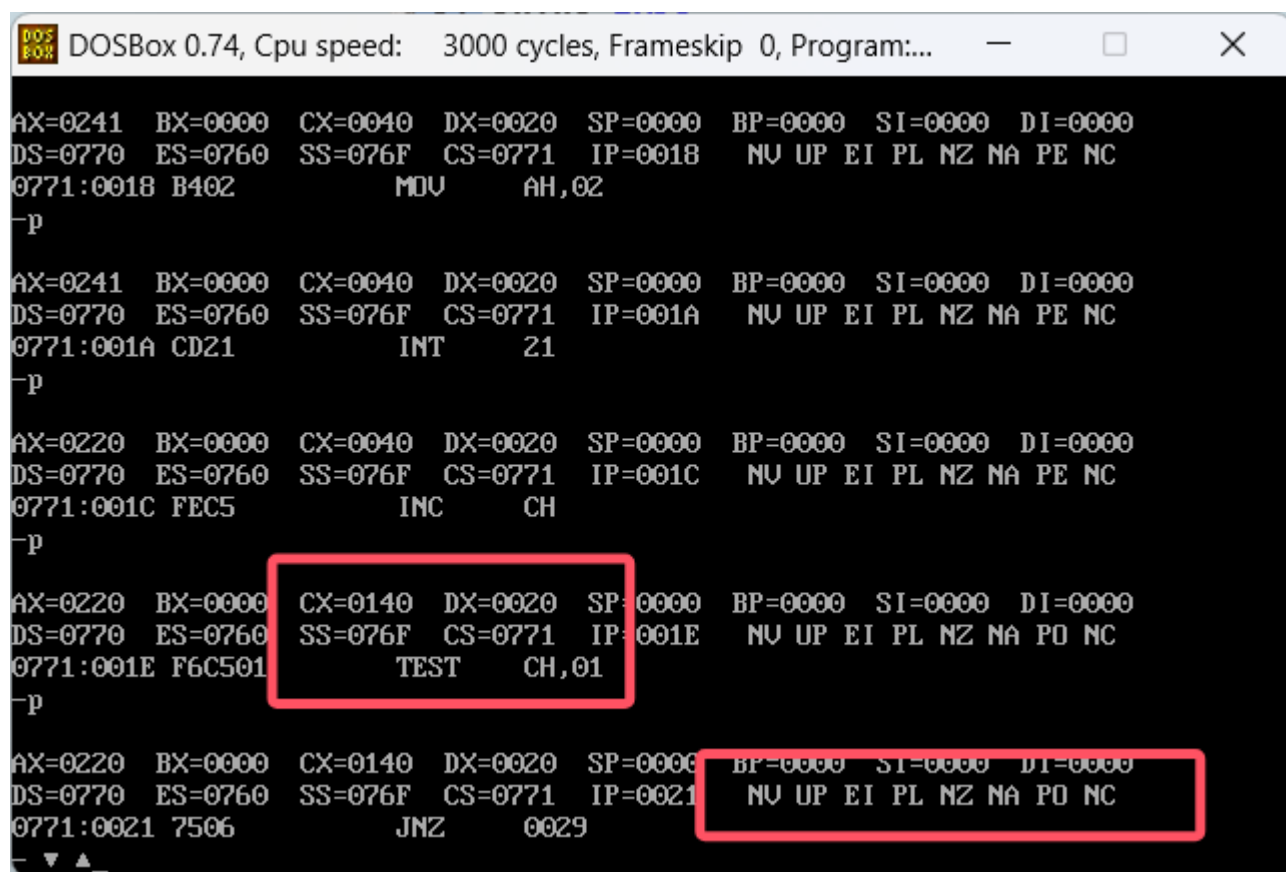
第二个字符为空格，空格的ascii码大于Z，不需要调整，直接跳转到next函数

实验2的思路为：

循环中依次遍历每个十六进制，因为要实现每两个输出一个换行符，可以定义ch=0，每次加1，通过test指令判断ch是否为偶数，如果是偶数，就输出一个换行符，如果不是，就直接跳转到next函数

```
INC CH
TEST CH,01H
JNZ NEXT
MOV DL,0AH
MOV AH,02H
INT 21H
```

当ch为1时，test最后一位，nz表示最后一位为1，则不输出换行符



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...
AX=0241 BX=0000 CX=0040 DX=0020 SP=0000 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0018  NU UP EI PL NZ NA PE NC
0771:0018 B402          MOV     AH,02
-p
AX=0241 BX=0000 CX=0040 DX=0020 SP=0000 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=001A  NU UP EI PL NZ NA PE NC
0771:001A CD21          INT     21
-p
AX=0220 BX=0000 CX=0040 DX=0020 SP=0000 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=001C  NU UP EI PL NZ NA PE NC
0771:001C FEC5          INC     CH
-p
AX=0220 BX=0000 CX=0140 DX=0020 SP=0000 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=001E  NU UP EI PL NZ NA PO NC
0771:001E F6C501       TEST    CH,01
-p
AX=0220 BX=0000 CX=0140 DX=0020 SP=0000 BP=0000 SI=0000 DI=0000
DS=0770 ES=0760 SS=076F CS=0771 IP=0021  NU UP EI PL NZ NA PO NC
0771:0021 7506          JNZ    0029
```

当ch为2时，与最后一位test之后，ZR表示最后一位为0，表示目前为偶数，可以输出换行符。

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program:...  
AX=0242 BX=0000 CX=0140 DX=0020 SP=0000 BP=0000 SI=0001 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=001A  NU UP EI PL NZ NA PE NC  
0771:001A CD21          INT     21  
-p  
AX=0220 BX=0000 CX=0140 DX=0020 SP=0000 BP=0000 SI=0001 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=001C  NU UP EI PL NZ NA PE NC  
0771:001C FEC5          INC     CH  
-p  
AX=0220 BX=0000 CX=0240 DX=0020 SP=0000 BP=0000 SI=0001 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=001E  NU UP EI PL NZ NA PO NC  
0771:001E F6C501        TEST    CH,01  
-p  
AX=0220 BX=0000 CX=0240 DX=0020 SP=0000 BP=0000 SI=0001 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=0021  NU UP EI PL ZR NA PE NC  
0771:0021 7506          JNZ     0029  
-p  
AX=0220 BX=0000 CX=0240 DX=0020 SP=0000 BP=0000 SI=0001 DI=0000  
DS=0770 ES=0760 SS=076F CS=0771 IP=0023  NU UP EI PL ZR NA PE NC  
0771:0023 B20A          MOV     DL,0A
```

【总结或讨论】

通过这两个汇编语言编程任务，掌握了字符编码转换、循环控制和条件判断等关键编程技能。在实现大写字母转小写的过程中，加深了对ASCII码表的理解，并通过实际操作学会了字符转换技巧；而在处理十六进制数转ASCII码的任务中，锻炼了循环和条件判断语句的应用，提高了格式化输出的能力。这些实践不仅增强了对计算机基础概念的认识，也为解决更复杂的编程问题打下了坚实的基础。