

附件3

西安电子科技大学

微机原理与接口技术 课程大作业报告

名称 五种功能的上机实验

人工智能 学院 2120011 班

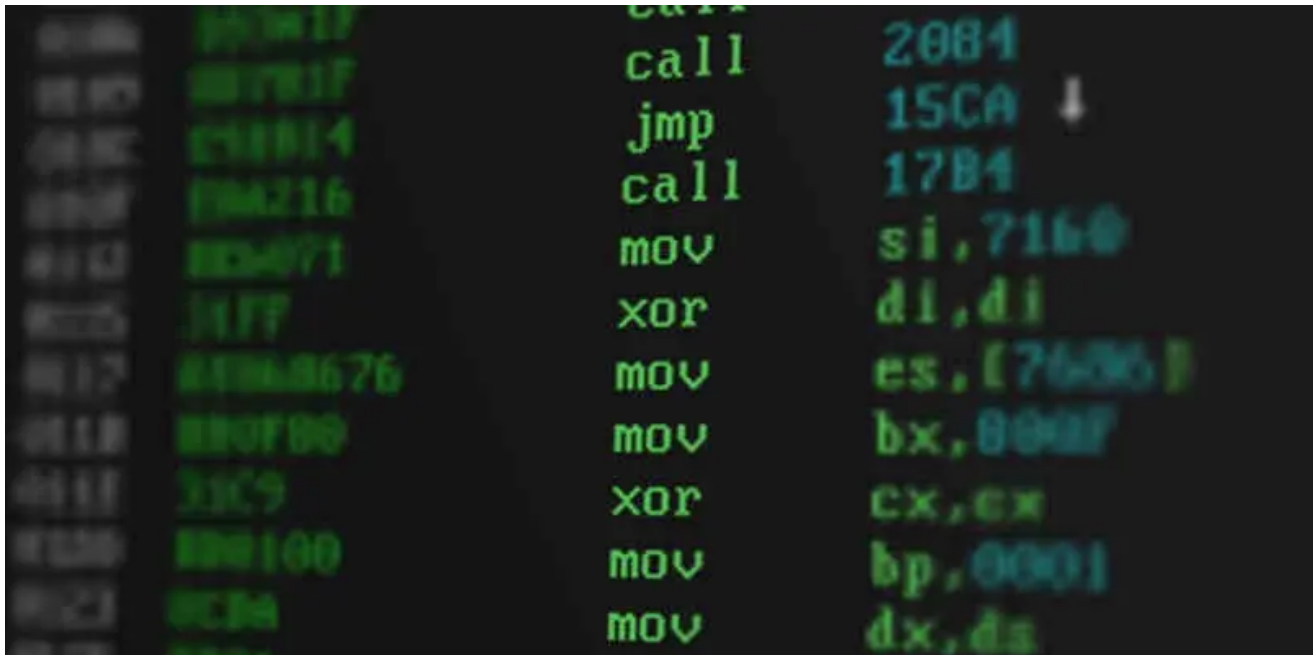
姓名 游霄童 学号 21009200158

同作者 无

日期 2023 年 10 月 26 日

成 绩

微机原理与接口技术大作业



一、实验要求

编写程序实现下列5项功能，通过从键盘输入1~5进行菜单式选择。

(1)按数字键“1”，完成将字符串中的小写字母变换成大写字母。用户输入由英文大小写字母或数字

0~9组成的字符串(以回车结束)，变换后按下列格式在屏幕上显示：

<原字符串>: abcdgyt0092

<新字符串>: ABCDGYT0092

按任意键重做；按Esc键返回主菜单。

(2)按数字键“2”，完成在字符串中找最大值。用户输入由英文大小写字母或数字0~9组成的字符串(以回车结束)，找出最大值后按下列格式在屏幕上显示：

<原字符串>The maximum is <最大值>.

按任意键重做；按Esc键返回主菜单。

(3)按数字键“3”，完成输入数据组的排序。用户输入一组十进制数值(小于255)，然后变换成十六进制数，并按递增方式进行排序，按下列格式在屏幕上显示：

<原数值串>

<新数值串>

按任意键重做；按Esc键返回主菜单。

(4)按数字键“4”，完成时间的显示。首先提示用户对时，即改变系统的定时器HH:MM:SS(以冒号

间隔，回车结束)，然后在屏幕的右上角实时显示出时间HH:MM:SS。

按任意键重新对时；按Esc键返回主菜单。

(5)按数字键“5”，结束程序的运行，返回操作系统。

二、基本原理及步骤

分模块进行各个框架功能的实现，首先完成主菜单的部分，可以选择进入各个功能模块，在按 **Esc**后也是回到此主菜单。各功能模块中使用子程序调用完成了功能的主要部分，其余则是调整输入位置的一些细节部分。

(1) 按数字键“1”，完成将字符串中的小写字母转换成大写字母。

- 接收用户输入的字符串，判断每个字符是否为小写字母，如果是则将其转换为大写字母。
- 将转换后的字符串显示在屏幕上。

步骤：

1. 初始化堆栈和寄存器。
2. 接收用户输入的字符串，存储在堆栈中。
3. 判断每个字符是否为小写字母，如果是则将其转换为大写字母。
4. 将转换后的字符串显示在屏幕上。
5. 清空堆栈和寄存器。

(2) 按数字键“2”，完成在字符串中找最大值。

- 接收用户输入的字符串，判断每个字符是否为数字，如果是则将其转换为数字。
- 比较每个数字，找到最大值。
- 将最大值显示在屏幕上。

步骤：

1. 初始化堆栈和寄存器。
2. 接收用户输入的字符串，存储在堆栈中。
3. 判断每个字符是否为数字，如果是则将其转换为数字。
4. 比较每个数字，找到最大值。
5. 将最大值显示在屏幕上。
6. 清空堆栈和寄存器。

(3) 按数字键“3”，完成输入数据组的排序。

- 接收用户输入的数据组，判断每个数据是否为十进制数值，如果是则将其转换为十六进制数。
- 将转换后的数据按递增方式进行排序。
- 将排序后的数据显示在屏幕上。

步骤：

1. 初始化堆栈和寄存器。
2. 接收用户输入的数据组，存储在堆栈中。
3. 接受一个包含十进制数字的数组作为输入，并将排序后的数组以十六进制格式输出。以下是该过程中涉及步骤的高层描述：
 1. 程序首先提示用户输入由空格分隔的一串十进制数字。
 2. 然后，它使用ChangeH过程将字符串中的每个数字转换为其对应的ASCII码。

3. 程序将转换后的ASCII码存储在一个名为Array的数组中。
4. 接着，程序使用冒泡排序算法对Array中的元素进行排序。
5. 排序完成后，程序将排序后的数组以十六进制格式显示在屏幕上。

需要注意的是，ChangeH过程用于将十进制数字转换为其对应的ASCII码，而Sort过程用于执行冒泡排序算法。

(4) 按数字键“4”，完成时间的显示。

- 提示用户对时，即改变系统的定时器HH:MM:SS。
- 在屏幕的右上角实时显示出时间HH:MM:SS。

步骤：

1. 初始化堆栈和寄存器。
2. 提示用户对时，即改变系统的定时器HH:MM:SS。
3. 在屏幕的右上角实时显示出时间HH:MM:SS。
4. 清空堆栈和寄存器。

(5) 按数字键“5”，结束程序的运行，返回操作系统。

- 结束程序的运行，返回操作系统。

步骤：

1. 初始化堆栈和寄存器。
2. 结束程序的运行，返回操作系统。
3. 清空堆栈和寄存器。

三、程序设计

分模块进行各个框架功能的实现，首先完成主菜单的部分，可以选择进入各个功能模块，在按Esc后也是回到此主菜单。各功能模块中使用子程序调用完成了功能的主要部分，其余则是调整输入位置的一些细节部分。

1、代码段和堆栈段

```
DATA SEGMENT
    ChooseModel DB 'Plesse input the number(1-5) to get in the
model:', '$'
    Model1txt1 DB 'Now, we are doing function 1:', '$'
    Model1txt2 DB 'input character:', '$'
    Model2txt1 DB 'Now, we are doing function 2:', '$'
    Model2txt2 DB 'input character:', '$'
    Model2txt3 DB 'max is:', '$'
    Model3txt1 DB 'Now, we are doing function 3:', '$'
    Model3txt2 DB 'input the decimal number:', '$'
    Model4txt1 DB 'Now, we are doing function 4:', '$'
    Model4txt2 DB 'press anykey to display the time', '$'
    Finaltxt DB 'What do you want to do next"Main Menu or Redo"[ESC or
any other key]:', '$'
```

```

TIPInputTIME DB 'Please input TIME(HH:MM:SS):','$'

STRING DB 100, DB 0, DB 100 DUP(0)
MaxStr DB 0
Array DB 100 DUP(0)
ArrayLength DB 0
SpiltStr DB 20H;空格的ASCII
Decimal DB 10

DATA ENDS

STACK SEGMENT STACK

STACK ENDS

```

其中诸多txt 是用来提示输入输出的。STRING 主要用于存放输入的字符串。MaxStr 则在第二个功能寻找最大值中使用到了。Array 在第三个功能中存放转为十进制后的数，ArrayLength 是这个数组的长度，SplitStr 存的是空格的 ASCII 码值，用来以空格分开字符串。Decimal 是十进制的意思，这里存放 10，在转换十进制时用到。

2、主菜单和各模块界面（不包含主要功能实现部分）

```

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

START:
    MOV AX,DATA
    MOV DS,AX
Menu:
    MOV AH,0
    MOV AL,3
    INT 10H
    ;设置显示器模式

    MOV AH,9
    LEA DX,ChooseModel
    INT 21H
    MOV AH,1
    INT 21H

    CMP AL,'1'
    JZ Model1
    CMP AL,'2'
    JZ Model2
    CMP AL,'3'

```

```
JZ Model3
CMP AL,'4'
JZ Model4
CMP AL,'5'
JZ Model5
```

AH=0, INT 10H 可以用来设置显示器模式, AL 寄存器表示欲设定的模式。

INT 21H 用来读入或者输出字符串。AH=01H 时, 键盘输入到 AL 中并显示在显示屏上。
AH=09H 时, 输入字符串 (DS:DX = 字符串首地址), 遇到'\$'结束。

```
Model1:

MOV AH,0
MOV AL,3
MOV BL,0
INT 10H
;清屏幕
MOV AH,2
MOV BH,0
MOV DL,0
MOV DH,0
INT 10H
;设置光标
MOV AH,9
LEA DX,Model1txt1
INT 21H
;显示Model1txt1
MOV AH,2
MOV BH,0
MOV DL,0
MOV DH,1
INT 10H
;设置光标
MOV AH,9
LEA DX,Model1txt2
INT 21H
;显示Model1txt2
CALL CapitalizeString
MOV AH,2
MOV BH,0
MOV DL,0
MOV DH,5
INT 10H
;设置光标
```

```

LEA DX,Finaltxt
MOV AH,9
INT 21H
;显示Finaltxt
MOV AH,8
INT 21H
CMP AL,1BH;Esc的ascii码值
JZ Menu
JMP Model1;检测键盘输入是否为Esc,若是则返回主菜单,否则重做

```

以 Model1 为例，各功能模块主界面主要是调整输入输出位置。AH=02H，INT 10H 可以设置光标位置。位置用 DH、DL 表示，DH 表示列号，DL 表示行号，BH 表示页号。

3、功能一的具体实现

程序主要使用遍历访问的方法，遍历字符串，当找到小写字母后把原字符与上 5FH 便可以得到其对应大写字母的 ASCII 码。

```

CHANGE:
    CMP BYTE PTR[SI],41H ;STRING为字符串,将其和 41H (大写 A) 比较
    JB INCMNUMBER        ;若小于 41H,就是数字,处理下一个,若为字母,与
    5FH加和,得到大写字母的 ASCII码。
    AND BYTE PTR[SI],5FH
INCMNUMBER:
    INC SI
    LOOP CHANGE

```

STRING 为字符串，将其和 41H（大写A）比较，若小于 41H，就是数字，处理下一个，若为字母，与 5FH 加和，得到大写字母的 ASCII 码。

4、功能二的具体实现

```

LEA SI,STRING+2
MOV CH,0
MOV CL,STRING+1
MOV DL,[SI]
MAX:
    CMP DL,[SI]
    JA INCMNUMBER2
    MOV DL,[SI]

```

```

INCNUMBER2:
    INC SI
LOOP MAX
;找最大值并放在DL中

MOV AH,2
INT 21H
;显示DL

```

从一个字符开始一次比较，若有更大的，更新其为最大值，直到遍历完整个字符串，最后将结果放到 DL 中，利用 AH=02H，INT21H 可直接显示 DL 值。

5、功能三的具体实现

```

LEA SI,STRING+2
LEA DI,Array
MOV DH,0      ;用于存放数组个数
MOV AL,0      ;用于存放该数字十进制
MOV DL,0      ;用于存放该数字位数
;SpiltStr中存储着空格，spilt函数是为了将以空格为间隙的输入数字读取 并转为十进制
数
Spilt:

    MOV BL,[SpiltStr]
    CMP [SI],BL
    JZ Addnum
    INC SI
    INC DL
    JMP NotSpilt
Addnum:
    INC DH

    PUSH CX
    PUSH SI

    MOV AL,DL
    MOV AH,0
    MOV CX,AX

Return:
    DEC SI
    LOOP Return

    MOV CX,AX
    DEC CX
    MOV AX,0
    MOV AL,[SI]

```



```
SUB AL, 30H;ASCII减30H就是十进制数
CMP CX, 0
JZ Continue
```

```
StrToDeci:
    INC SI
    MUL Decimal
    MOV BL, [SI]
    SUB BL, 30H
    ADD AL, BL
    LOOP StrToDeci
Continue:
    MOV [DI], AL
    INC DI
    MOV DL, 0

    POP SI
    INC SI;指向逗号下一个
    POP CX
NotSpilt:
    LOOP Spilt
```

遍历整个字符串，用空格将字符串分隔开，每遇到一个空格，用来存放数组长度的 DH+1，结束后在+1，DH 就是数组的长度。每一次 DL 用来存这个数的长度，如数 123 长度就为 3。

拿到这个数后，取这个数的每一位数字，数字的 ASCII 码减去 30H 就是数字本身。然后用下面的数学原理可以得到这个数原本的值。

数学原理：用 $x_1x_2x_3$ 表示一个数， x_1, x_2, x_3 分别为百位十位和个位的数字，那这个数的值可以表示为

$$((x_1 \times 10) + x_2) \times 10 + x_3$$

```
;给最后一个数再来一次
MOV AL, DL
MOV AH, 0
MOV CX, AX
```

```

Return2:
    DEC SI
    LOOP Return2

MOV CX, AX
DEC CX
MOV AX, 0
MOV AL, [SI]
SUB AL, 30H; ASCII减30H就是十进制数
CMP CX, 0
JZ Continue2

StrToDeci2:
    INC SI
    MUL Decimal
    MOV BL, [SI]
    SUB BL, 30H
    ADD AL, BL
    LOOP StrToDeci2

Continue2:
MOV [DI], AL
INC DI
MOV DL, 0

INC DH
MOV [ArrayLength], DH
;至此，已经将数字存入Array中

```

上述写法会在最后一次循环时不处理最后一个数，所以多写一遍用来处理最后一个数即可。

```

;输出16进制
;接下来输出16进制数
MOV AH, 2
MOV BH, 0
MOV DL, 0
MOV DH, 5
INT 10H
;设置光标
DispArray:
    MOV AL, [ArrayLength]
    MOV AH, 0
    MOV CX, AX
    LEA SI, Array

    DispOne:

```

```

MOV AL,[SI]
PUSH AX
SHR AL,4
CALL ChangeH
MOV AH,02
MOV DL,AL
INT 21H
POP AX
AND AL,0FH
CALL ChangeH
MOV AH,02
MOV DL,AL
INT 21H
MOV DL,48H
INT 21H
MOV DL,20H
INT 21H
INC SI

LOOP DispOne

```

```

ChangeH PROC NEAR
    CMP AL,10
    JNGE Change1
    ADD AL,7
Change1:
    ADD AL,30H
    RET
ChangeH ENDP

```

参考课本例 4.49 完成了以十六进制数输出的内容。一个数占据了 8 位，也就是两位十六进制数，用 SHR 指令每次取出 4 位（一位十六进制数）去处理。

;接下来是排序

```

MOV AL,[ArrayLength]
MOV CH,0
MOV CL,AL
DEC CX;外循环次数
LEA SI,Array
ADD SI,CX
LP1:
    PUSH CX
    PUSH SI

```

```

LP2:
    MOV AL,[SI]
    CMP AL,[SI-1]
    JAE NOXCHG
    XCHG AL,[SI-1]
    MOV [SI],AL
NOXCHG:
    DEC SI
    LOOP LP2
    POP SI
    POP CX
    LOOP LP1

```

参考课本例 4.43，采用双重循环使用冒泡法完成了数组的排序。输出同之前的输出

6、功能四的实现

```

TIME PROC NEAR

;2CH号功能获取时间并存储在Array中
MOV AH,2CH
INT 21H
MOV [Array],CH
MOV [Array+1],CL
MOV [Array+2],DH

MOV AH,2
MOV BH,0
MOV DL,0
MOV DH,2
INT 10H
;设置光标
LEA SI,Array
CALL DispDec

MOV DL,':'
MOV AH,02H
INT 21H
;时、分、秒添加间隔“: ”
INC SI
CALL DispDec

MOV DL,':'
MOV AH,02H
INT 21H

INC SI

```

```
CALL DispDec

RET

TIME ENDP
```

用 AH=2CH, INT21H 可以获取系统时间, CH:CL=时:分, DH:DL=秒:1/100 秒然后将时间存在 Array 中, 再逐一输出Array 中的每一个数, 中间用: 隔开。

```
DispDec PROC NEAR
    ;将获取的时间转化为10进制数
    MOV AL, [SI]
    CBW
    DIV Decimal
    ADD AH, '0'
    MOV BH, AH
    ADD AL, '0'

    MOV DL, AL
    MOV AH, 02H
    INT 21H

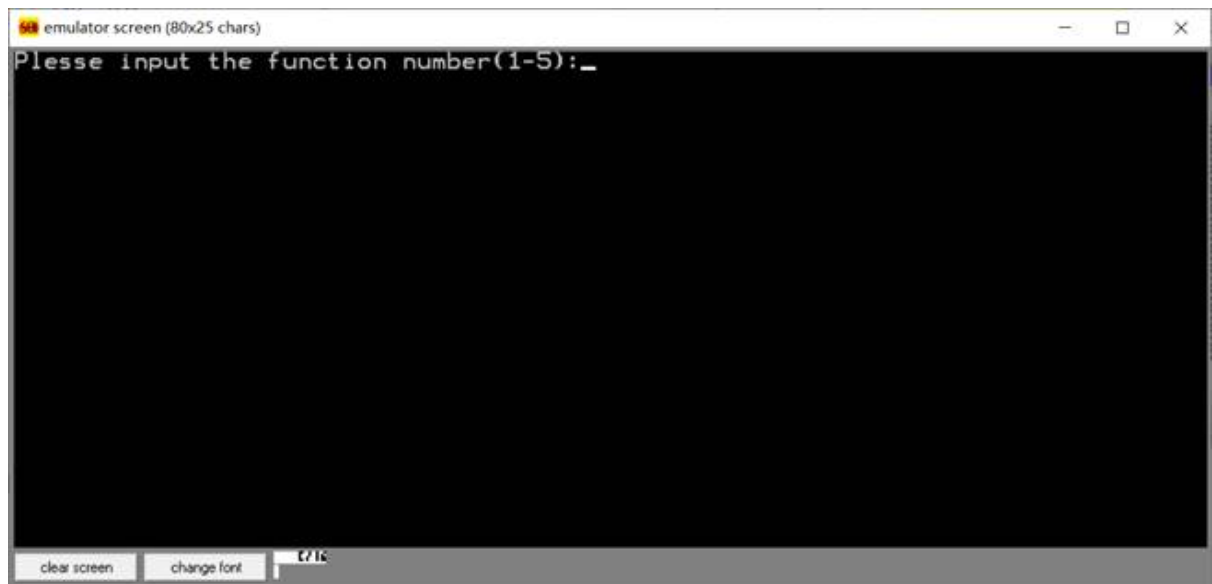
    MOV DL, BH
    INT 21H

    RET
DispDec ENDP
```

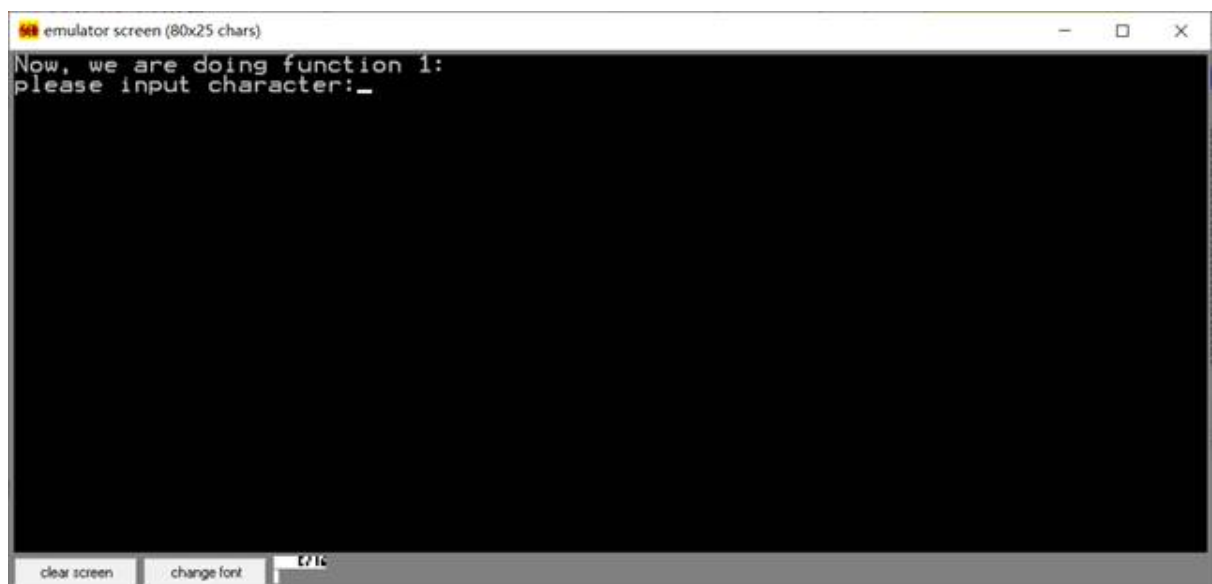
输出的时候需要将获取的时间转化为十进制数。

四、结果与分析

请求输入数字 1-5 来选择不同功能



输入 1:



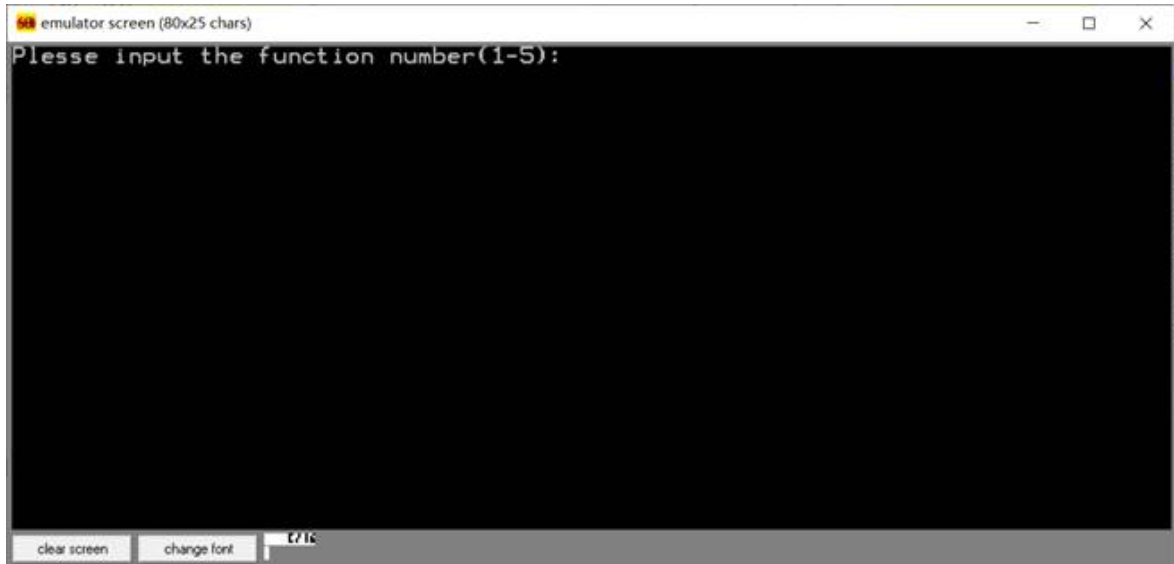
输入字符串 XidianUniv123



按出 Esc 外的其他键重做:



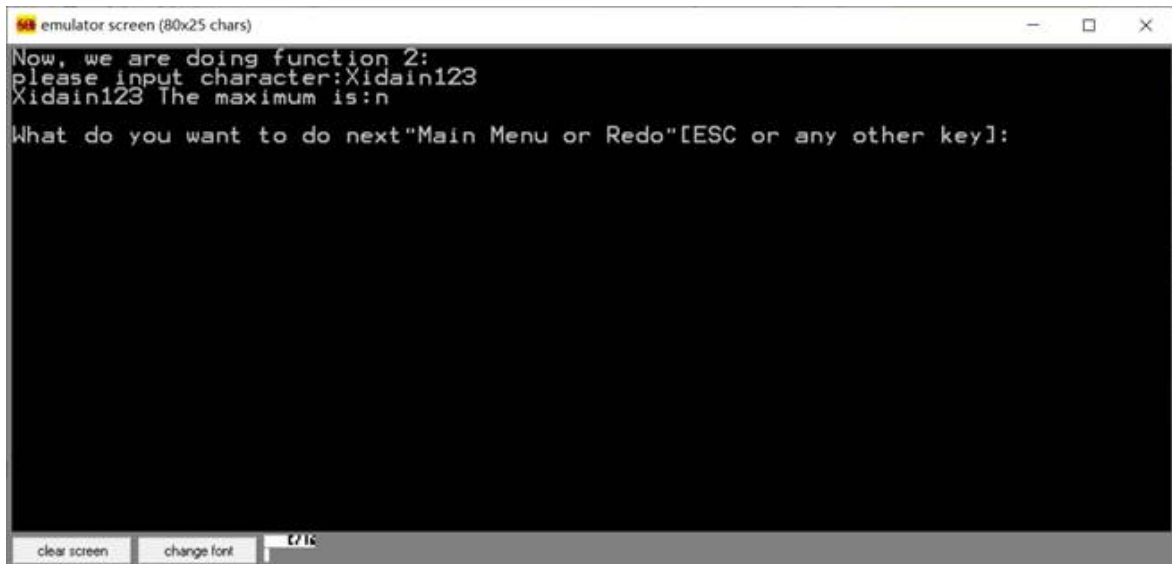
按 Esc 退出到主菜单进行选择：



输入 2:

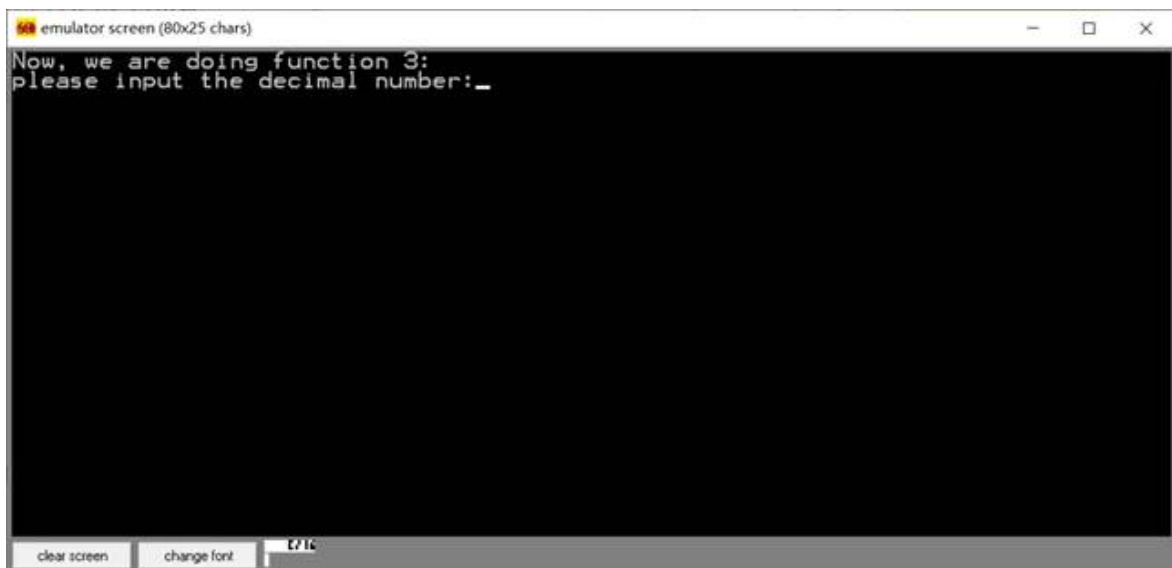


输入字符串: Xidian123



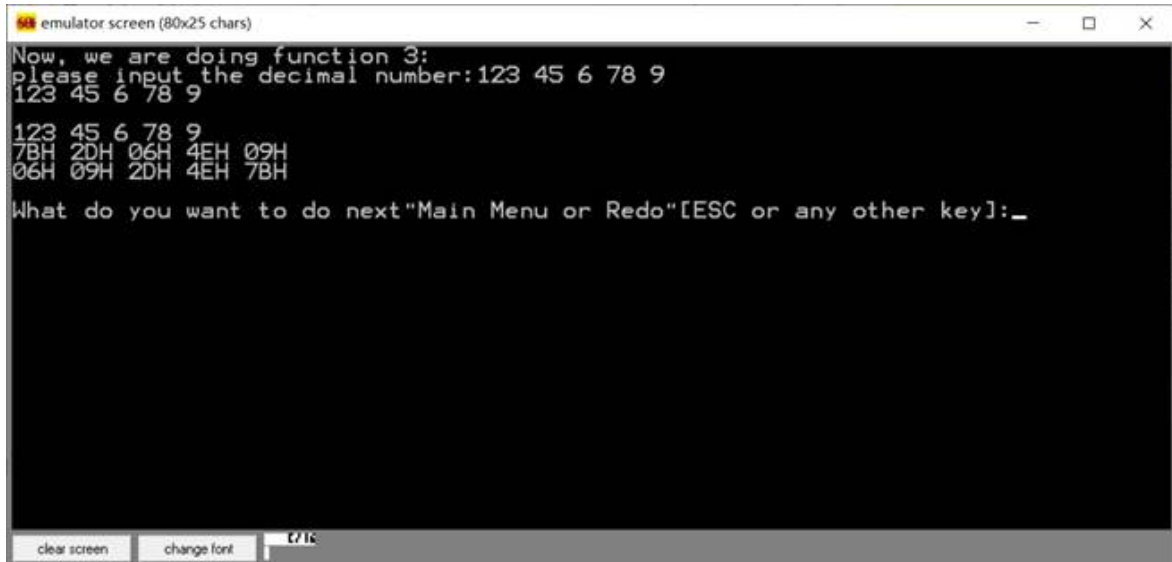
The screenshot shows a terminal window titled "emulator screen (80x25 chars)". The text inside the terminal reads: "Now, we are doing function 2:", "please input character:Xidain123", "Xidain123 The maximum is:n", and "What do you want to do next\"Main Menu or Redo\"[ESC or any other key]:". At the bottom of the terminal, there are two buttons labeled "clear screen" and "change font", and a small status bar showing "E716".

按 Esc 后输入 3:



The screenshot shows a terminal window titled "emulator screen (80x25 chars)". The text inside the terminal reads: "Now, we are doing function 3:", "please input the decimal number:_". At the bottom of the terminal, there are two buttons labeled "clear screen" and "change font", and a small status bar showing "E716".

输入一组数：123 45 6 78 9

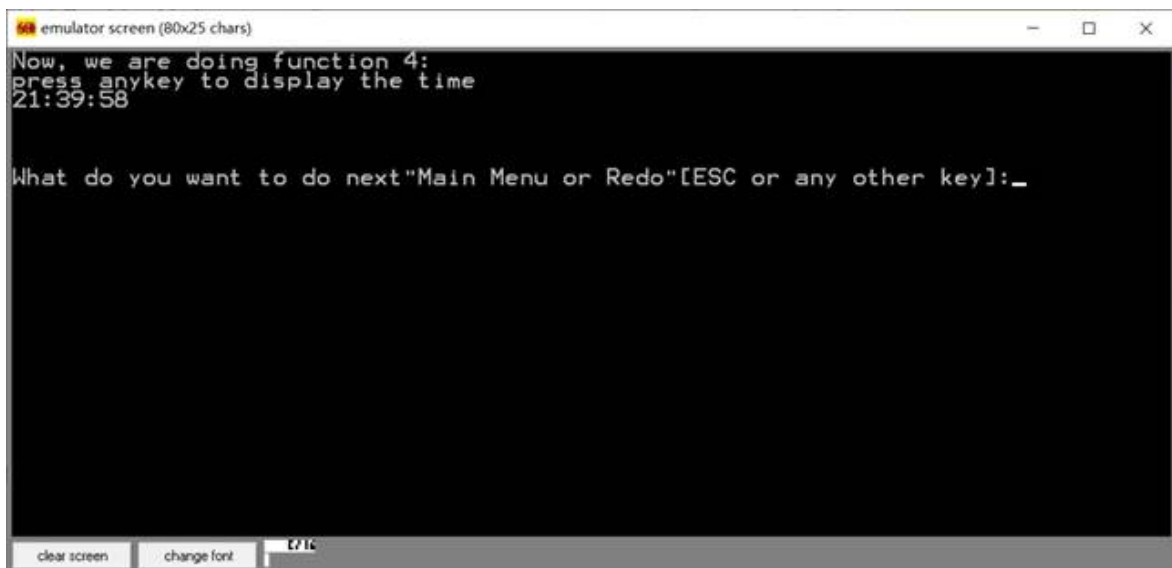


The screenshot shows a terminal window titled "emulator screen (80x25 chars)". The text inside the terminal is as follows:

```
Now, we are doing function 3:  
please input the decimal number:123 45 6 78 9  
123 45 6 78 9  
7BH 2DH 06H 4EH 09H  
06H 09H 2DH 4EH 7BH  
What do you want to do next"Main Menu or Redo"[ESC or any other key]:_
```

At the bottom of the terminal window, there are two buttons labeled "clear screen" and "change font", and a small status bar showing "E716".

按 Esc 后输入 4：显示时间

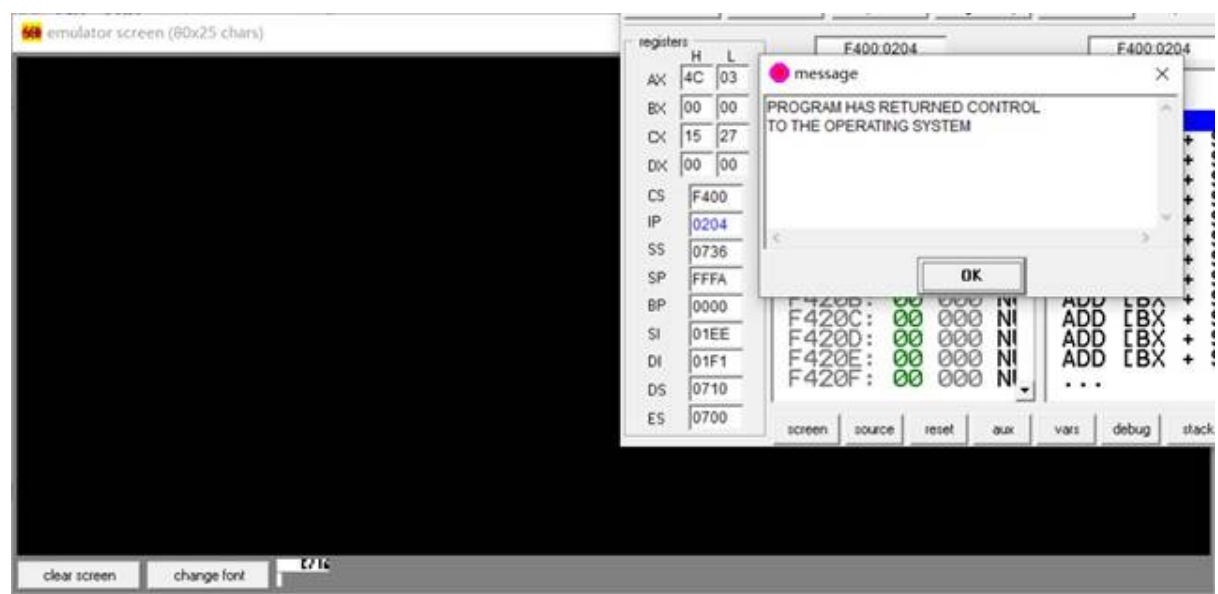


The screenshot shows a terminal window titled "emulator screen (80x25 chars)". The text inside the terminal is as follows:

```
Now, we are doing function 4:  
press anykey to display the time  
21:39:58  
What do you want to do next"Main Menu or Redo"[ESC or any other key]:_
```

At the bottom of the terminal window, there are two buttons labeled "clear screen" and "change font", and a small status bar showing "E716".

按 Esc 后输入 5：结束程序运行



五、心得体会

编程技能提升：通过实现不同的功能，有机会加深对汇编语言的理解，掌握不同功能的实现方法，提高了编程技能。**逻辑思维训练：**在处理菜单选择和功能实现时，需要具备良好的逻辑思维能力，确保每个功能按照要求正确执行，这有助于培养逻辑思维和问题解决能力。**用户交互设计：**这个实验涉及用户输入和输出的处理，锻炼了设计友好用户界面的能力，考虑如何清晰地提示用户操作，如何在屏幕上合适地显示信息。**调试和错误处理：**在编写程序的过程中，可能遇到了一些错误，学会了如何通过调试找到错误的原因并加以修复。