



西安电子科技大学
XIDIAN UNIVERSITY

微机原理实验课

人工智能学院

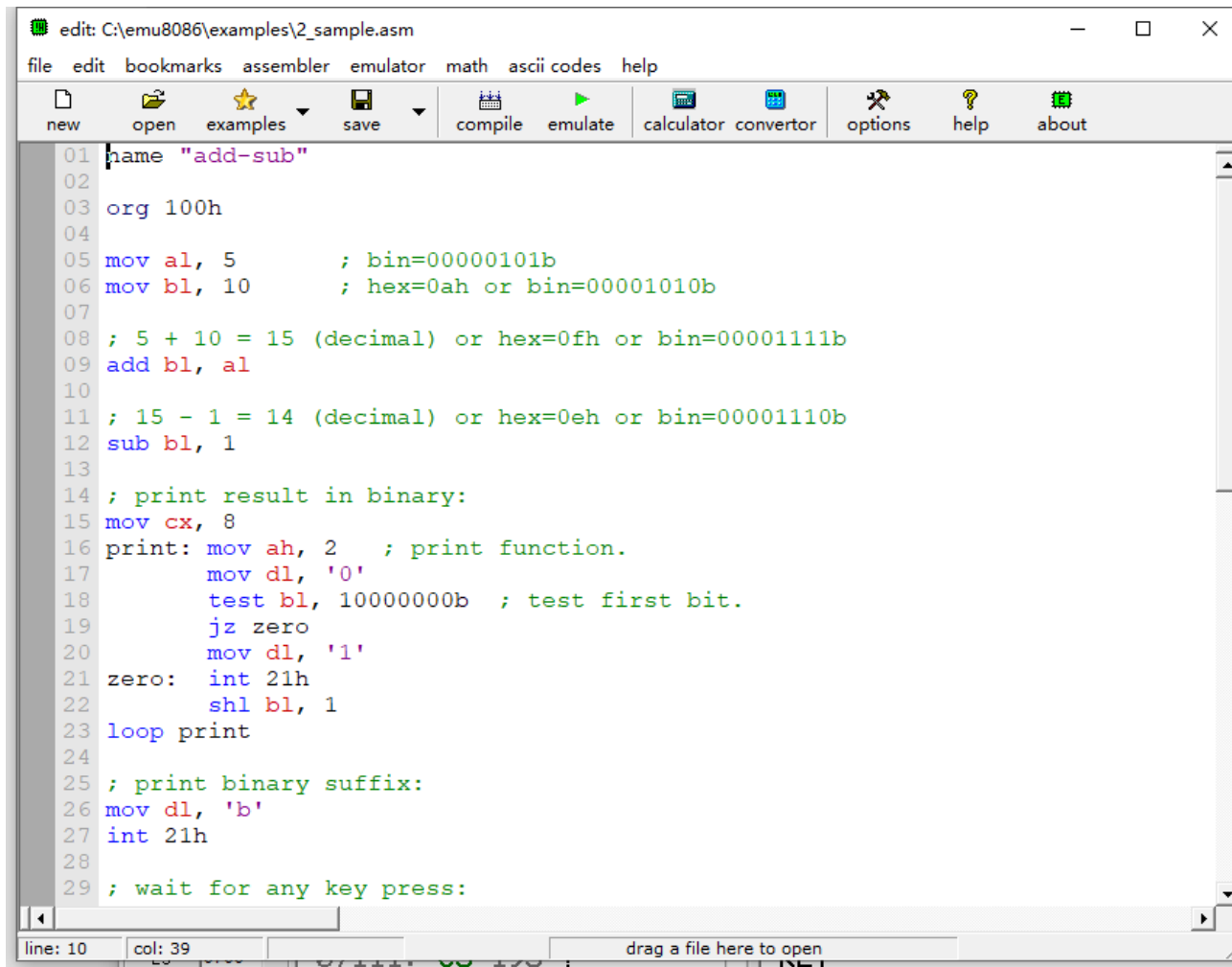
赵庆行

2021年10月28日



- 编程环境介绍
- 上机题分析
- 作业要求

emu8086使用

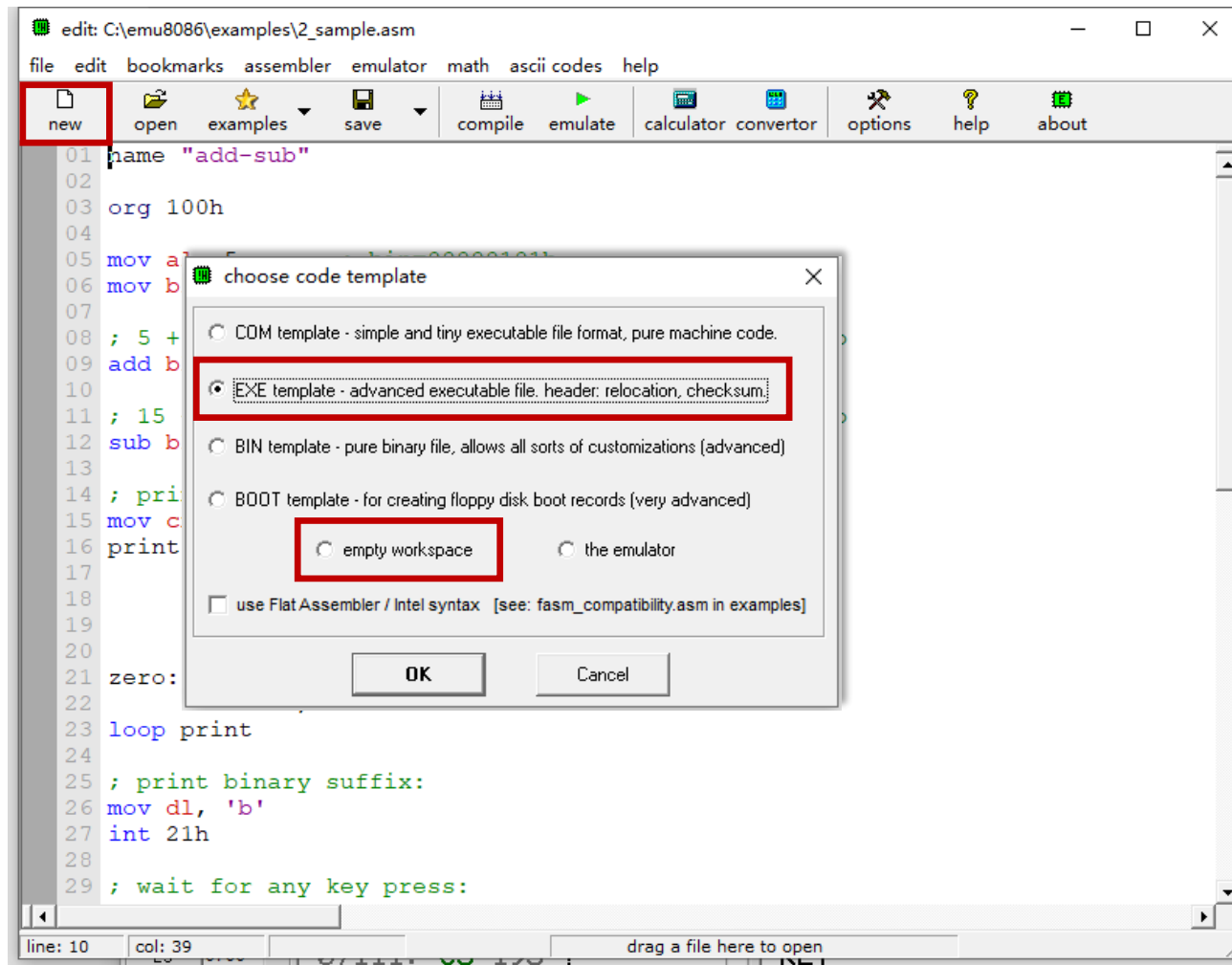


The screenshot shows the emu8086 software window with the following assembly code:

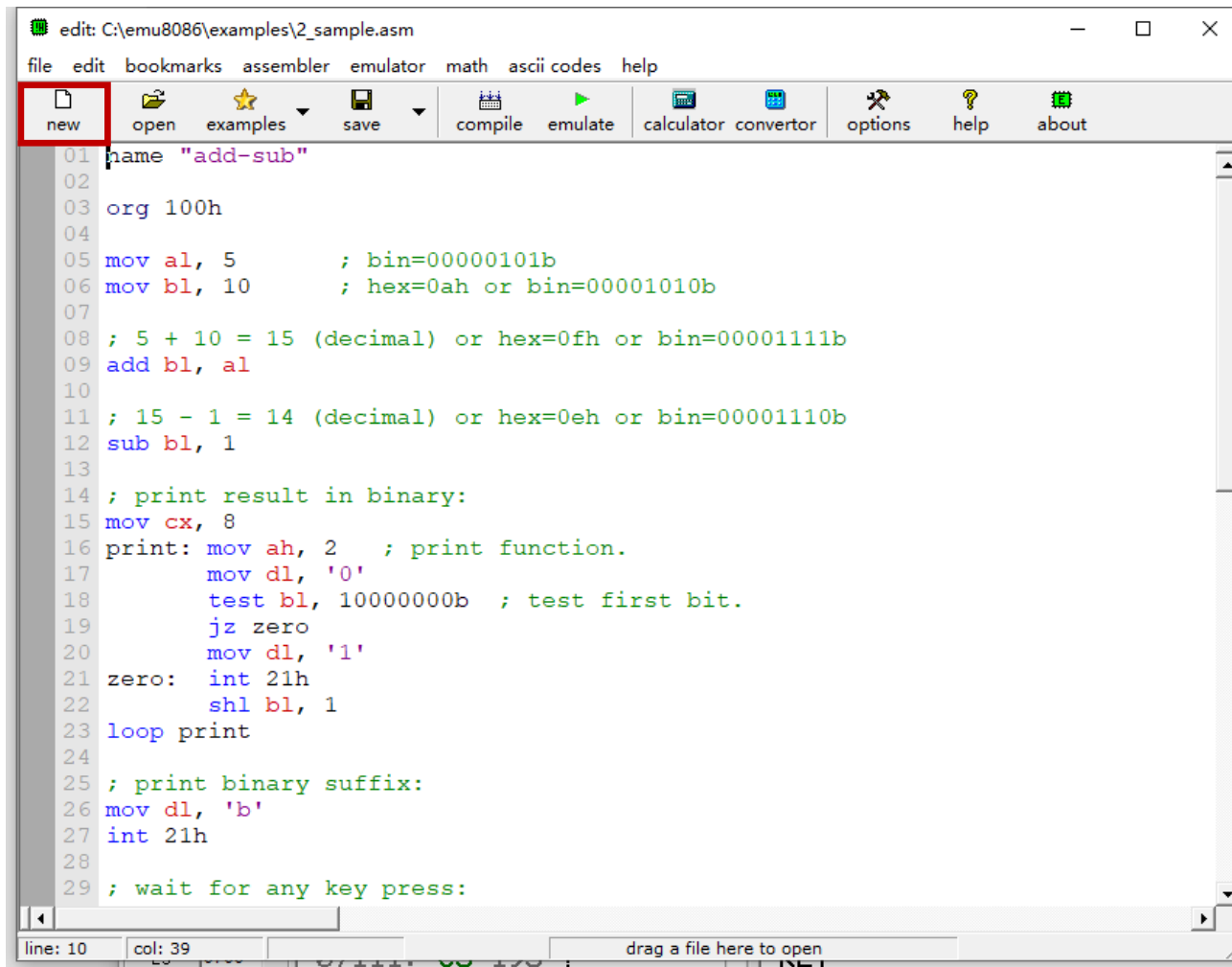
```
01 name "add-sub"
02
03 org 100h
04
05 mov al, 5      ; bin=00000101b
06 mov bl, 10     ; hex=0ah or bin=00001010b
07
08 ; 5 + 10 = 15 (decimal) or hex=0fh or bin=00001111b
09 add bl, al
10
11 ; 15 - 1 = 14 (decimal) or hex=0eh or bin=00001110b
12 sub bl, 1
13
14 ; print result in binary:
15 mov cx, 8
16 print: mov ah, 2 ; print function.
17         mov dl, '0'
18         test bl, 10000000b ; test first bit.
19         jz zero
20         mov dl, '1'
21 zero:   int 21h
22         shl bl, 1
23 loop print
24
25 ; print binary suffix:
26 mov dl, 'b'
27 int 21h
28
29 ; wait for any key press:
```

The status bar at the bottom indicates "line: 10" and "col: 39".

emu8086使用



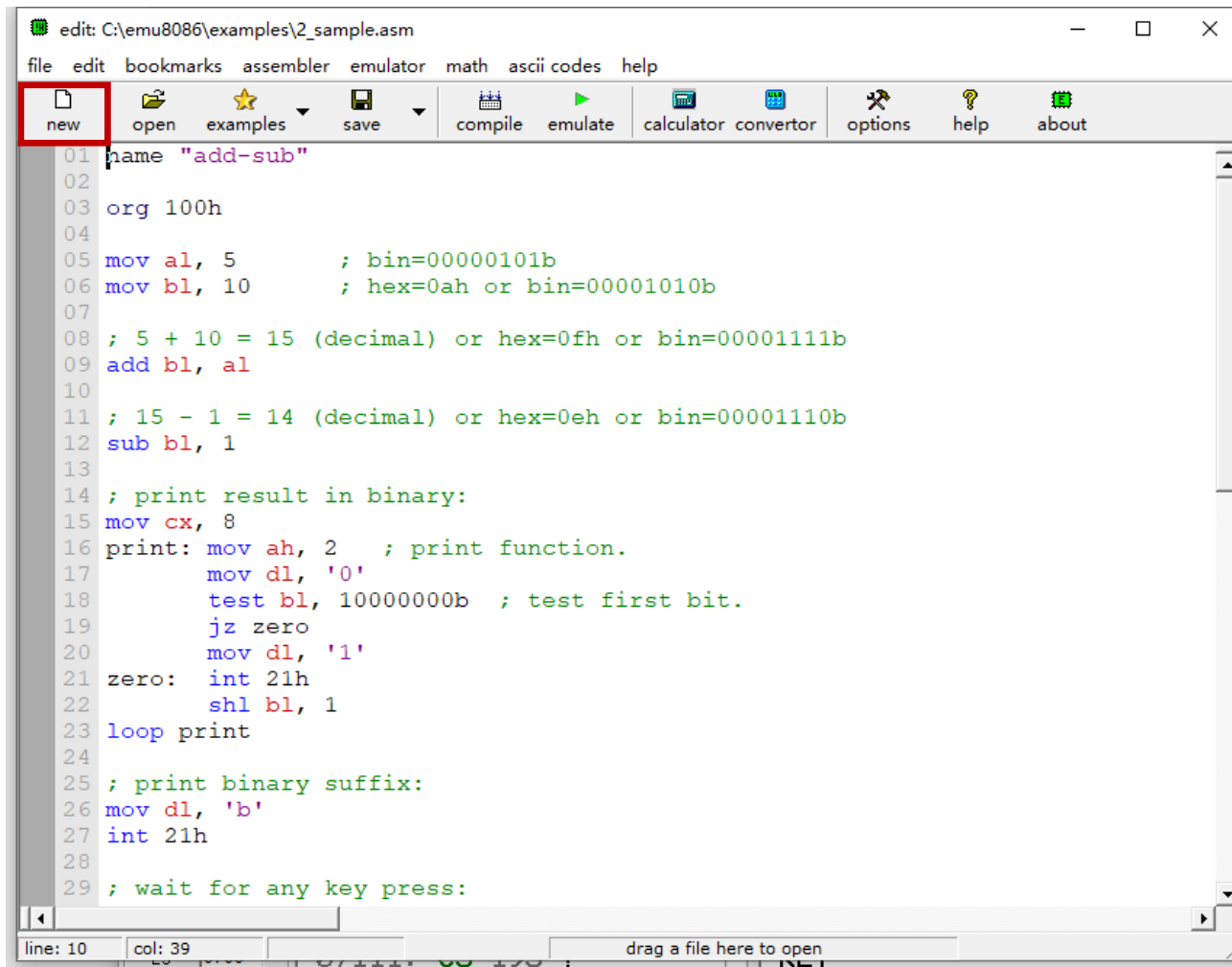
emu8086使用



```
edit: C:\emu8086\examples\2_sample.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 name "add-sub"
02
03 org 100h
04
05 mov al, 5      ; bin=00000101b
06 mov bl, 10     ; hex=0ah or bin=00001010b
07
08 ; 5 + 10 = 15 (decimal) or hex=0fh or bin=00001111b
09 add bl, al
10
11 ; 15 - 1 = 14 (decimal) or hex=0eh or bin=00001110b
12 sub bl, 1
13
14 ; print result in binary:
15 mov cx, 8
16 print: mov ah, 2      ; print function.
17         mov dl, '0'
18         test bl, 10000000b ; test first bit.
19         jz zero
20         mov dl, '1'
21 zero:   int 21h
22         shl bl, 1
23 loop print
24
25 ; print binary suffix:
26 mov dl, 'b'
27 int 21h
28
29 ; wait for any key press:
```

line: 10 col: 39 drag a file here to open

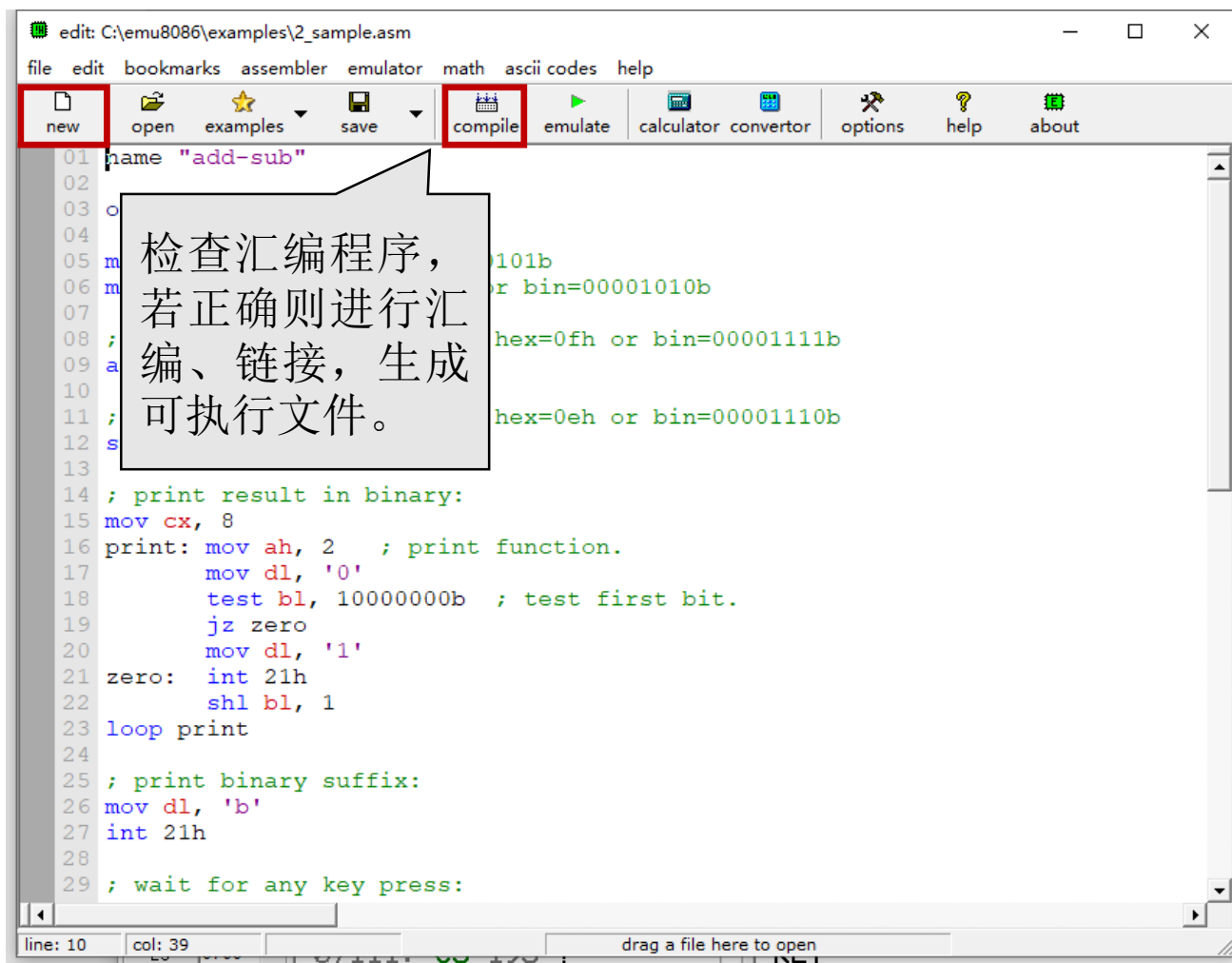
emu8086使用



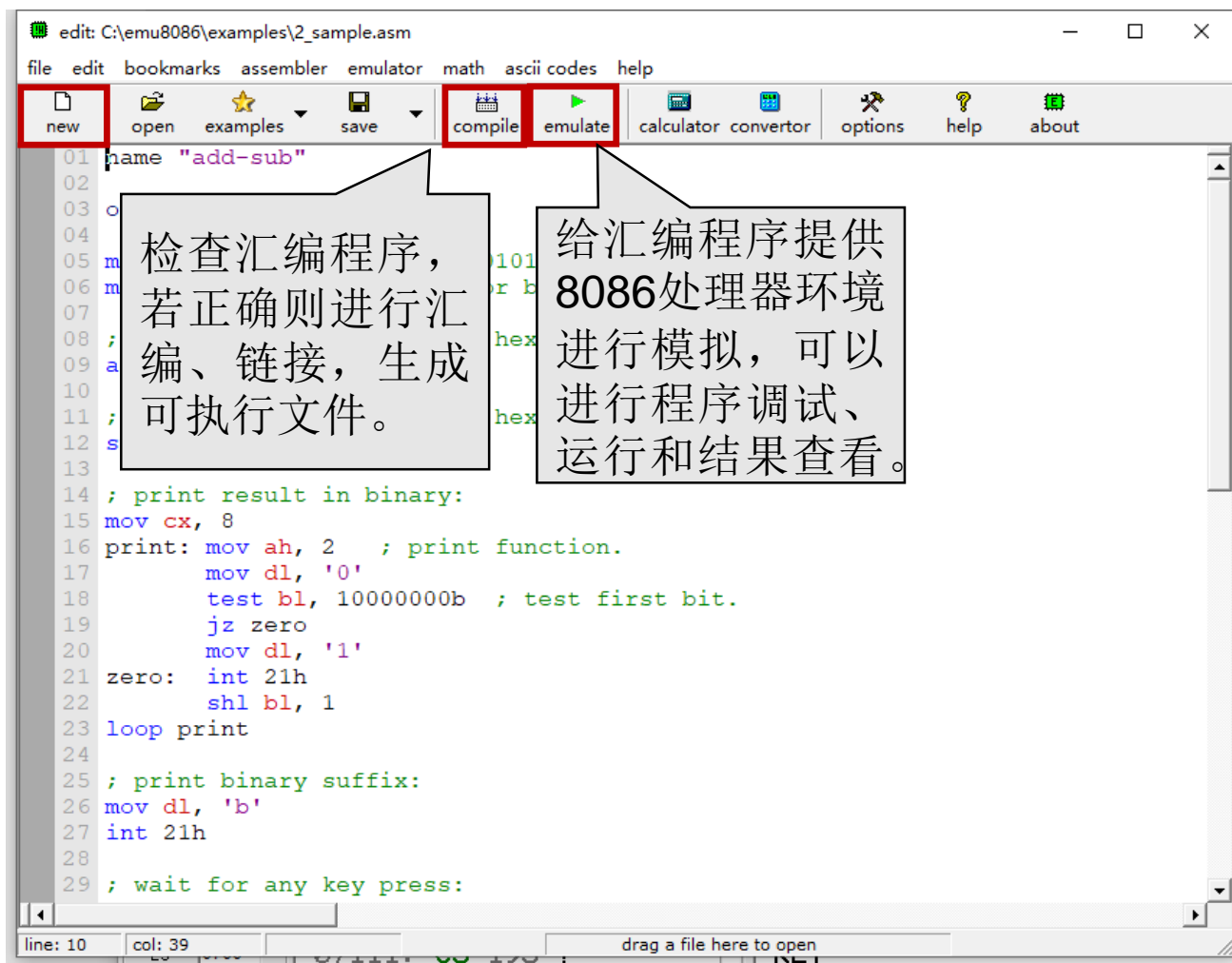
```
edit: C:\emu8086\examples\2_sample.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
01 name "add-sub"
02
03 org 100h
04
05 mov al, 5      ; bin=00000101b
06 mov bl, 10     ; hex=0ah or bin=00001010b
07
08 ; 5 + 10 = 15 (decimal) or hex=0fh or bin=00001111b
09 add bl, al
10
11 ; 15 - 1 = 14 (decimal) or hex=0eh or bin=00001110b
12 sub bl, 1
13
14 ; print result in binary:
15 mov cx, 8
16 print: mov ah, 2      ; print function.
17         mov dl, '0'
18         test bl, 10000000b ; test first bit.
19         jz zero
20         mov dl, '1'
21 zero:   int 21h
22         shl bl, 1
23 loop print
24
25 ; print binary suffix:
26 mov dl, 'b'
27 int 21h
28
29 ; wait for any key press:
```

line: 10 col: 39 drag a file here to open

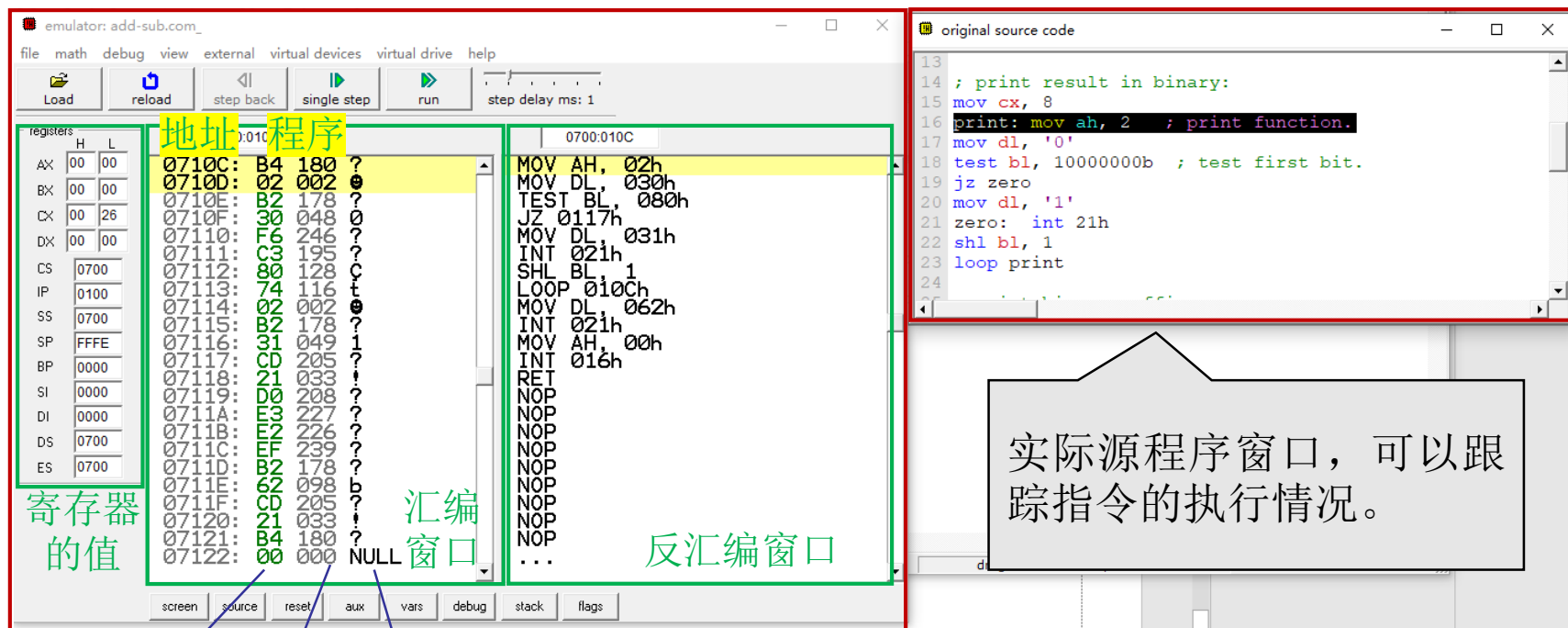
emu8086使用



emu8086使用

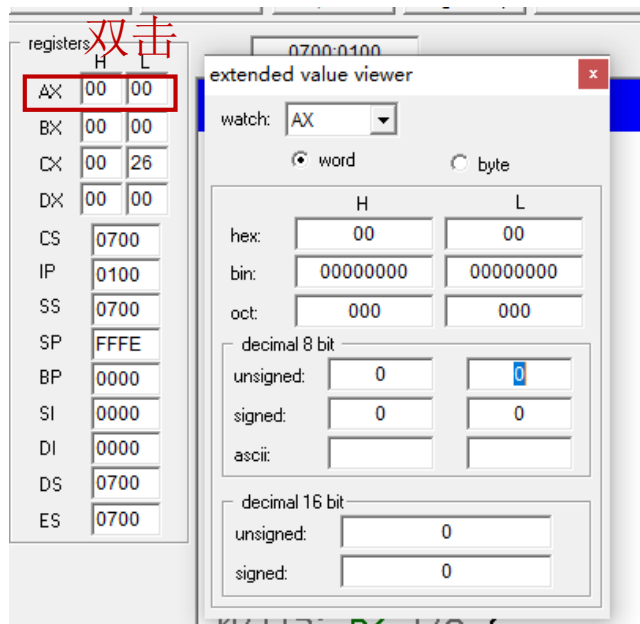


emu8086使用

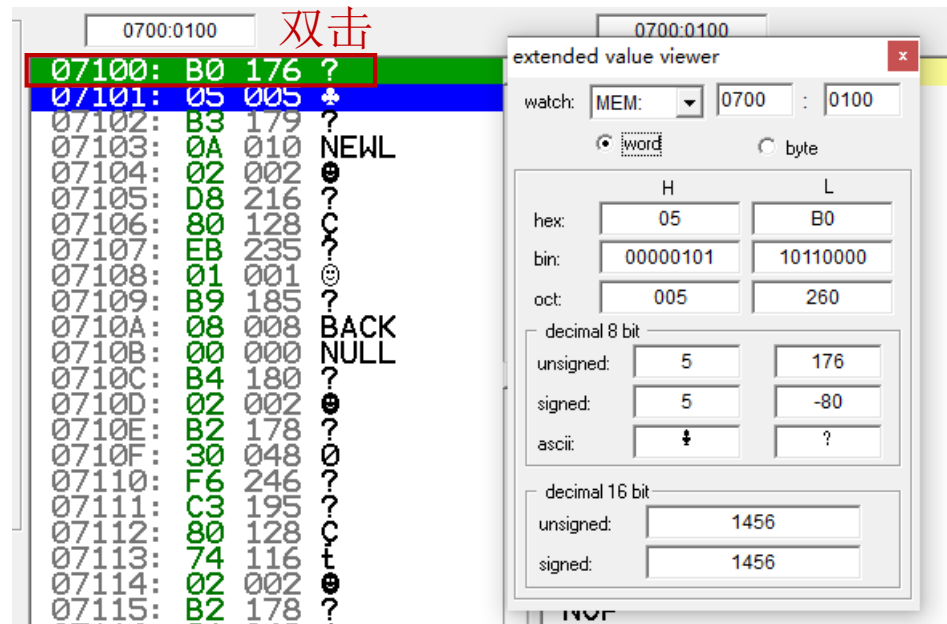


调试、运行窗口，可以选择执行或调试程序，查看寄存器、内存的内容，进行汇编和反汇编。

emu8086使用



修改寄存器的值



修改内存值

- 设置断点
- 条件停止

演示

(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>例如:abcdgyt0092

<新字符串> ABCDGYT0092

按任一键重做；按 Esc 键返回主菜单。

```
please input the function number (1~5): 1
```

```
Now, we are doing function 1:
```

```
please input character: XidianUniv123
```

```
XidianUniv123
```

```
XIDIANUNIV123
```

```
What do you want to do next" Main Menu or Redo" [ESC/any other key]: _
```

(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>例如:abcdgyt0092

<新字符串> ABCDGYT0092

按任一键重做；按 Esc 键返回主菜单。

```
please input the function number (1~5): 1
```

```
Now, we are doing function 1:
```

```
please input character: XidianUniv123
```

```
XidianUniv123
```

```
XIDIANUNIV123
```

```
What do you want to do next" Main Menu or Redo" [ESC/any other key]: _
```

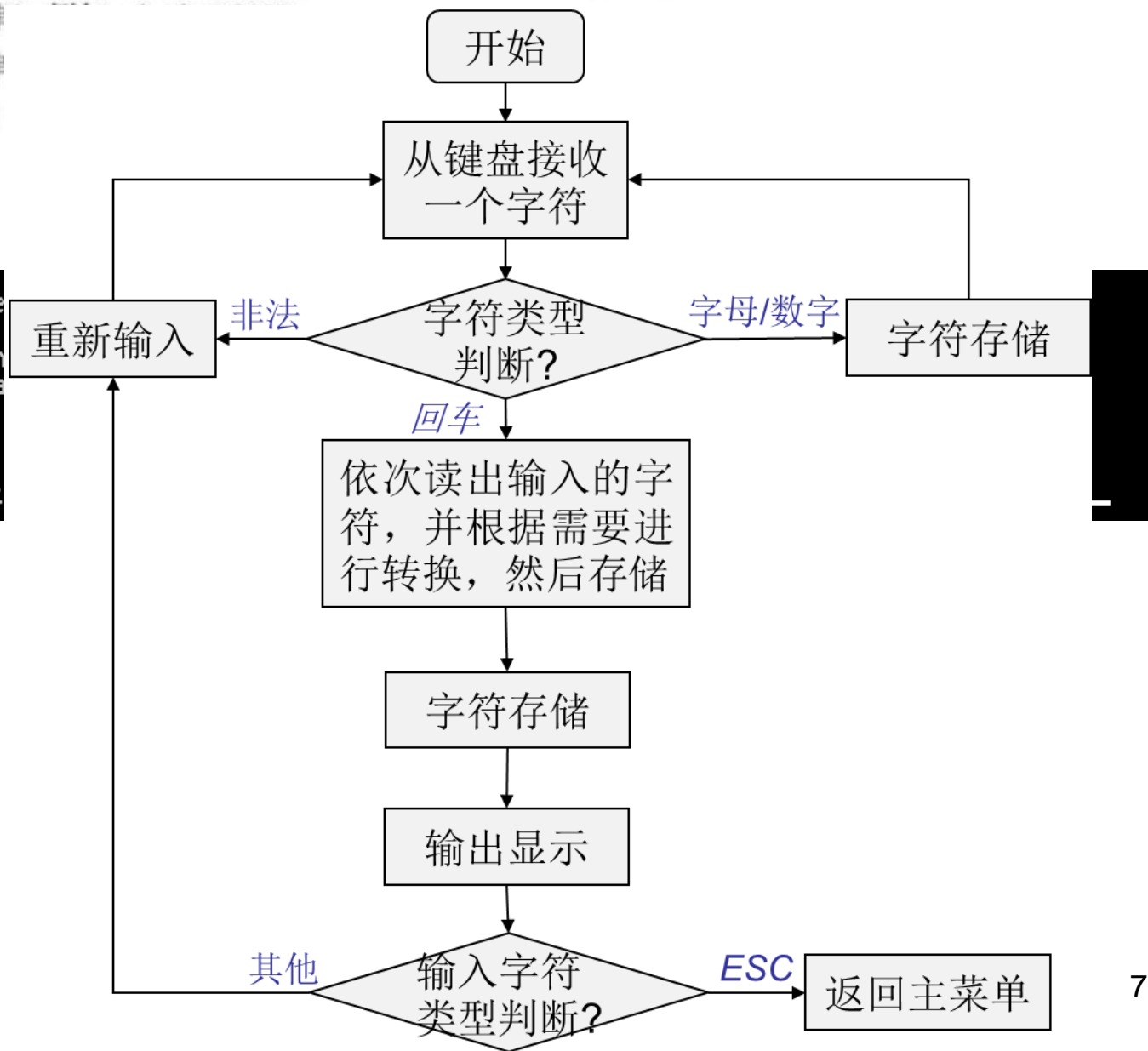
(1) 按数字键“1”，完成将字符串中的小写字母变换成大写字母。用户输入由英文大小写字母或数字0~9组成的字符串（以回车结束），变换后按下列格式在屏幕上显示：

<原字符串

<新字符串

按任一键重做：

```
please input the
Now, we are doin
please input cha
XidianUniv123
XIDIANUNIV123
What do you want
```



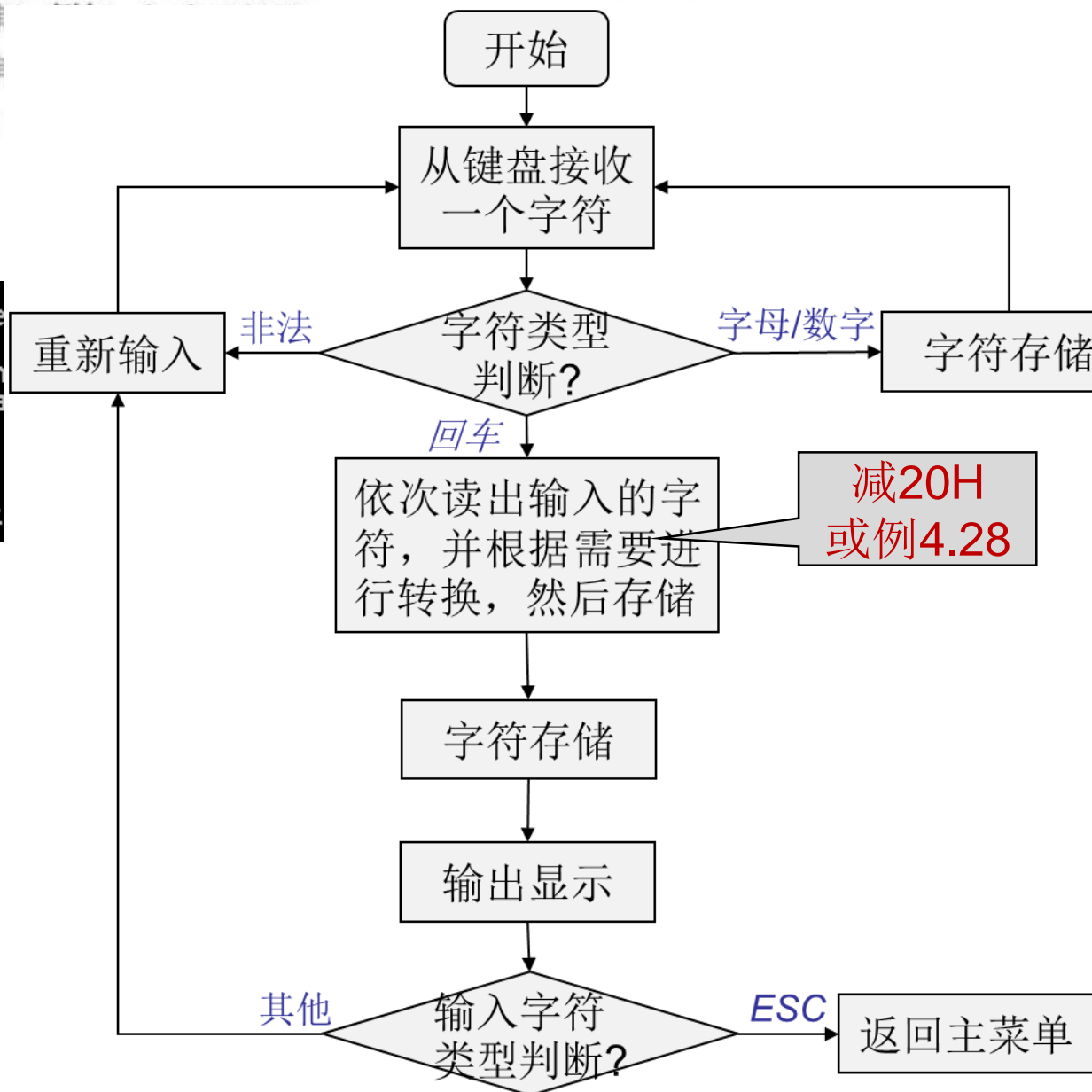
(1) 按数字键“1”，完成将字符串中的小写字母变换成大写字母。用户输入由英文大小写字母或数字0~9组成的字符串（以回车结束），变换后按下列格式在屏幕上显示：

<原字符串

<新字符串

按任一键重做：

```
please input the
Now, we are doin
please input cha
XidianUniv123
XIDIANUNIV123
What do you want
```



(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>.

按任一键重做；按 Esc 键返回主菜单。

```
please input the function number (1~5): 2
Now, we are doing function 2:
please input character: afd123
afd123 The maximum is: f
What do you want to do next" Main Menu or Redo" [ESC/any other key]:
```


(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>.

按任一键重做；按 Esc 键返回主菜单。

```
please input the function number (1~5): 2
Now, we are doing function 2:
please input character: afd123
afd123 The maximum is: f
What do you want to do next" Main Menu or Redo" [ESC/any other key]:
```

- 思路1：1) 输入的字符进行合法性检查，然后进行存储；2) 将存储的字符依次取出，然后与最大字符进行比较，并保留较大字符；3) 输出。

(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>.

按任一键重做；按 Esc 键返回主菜单。

```
please input the function number (1~5): 2
Now, we are doing function 2:
please input character: afd123
afd123 The maximum is: f
What do you want to do next" Main Menu or Redo" [ESC/any other key]:
```

- 思路1：1) 输入的字符进行合法性检查，然后进行存储；2) 将存储的字符依次取出，然后与最大字符进行比较，并保留较大字符；3) 输出。
- 思路2：1) 输入的字符进行合法性检查，并与（某寄存器中的）最大字符进行比较，保留两者的最大值，然后存储输入字符；2) 输出输入的字符串和最大值。

(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>.

按任一键重做；按 Esc 键返回主菜单。

```
please input the function number (1~5): 2
Now, we are doing function 2:
please input character: afd123
afd123 The maximum is: f
What do you want to do next" Main Menu or Redo" [ESC/any other key]:
```

- 思路1：1) 输入的字符进行合法性检查，然后进行存储；2) 将存储的字符依次取出，然后与最大字符进行比较，并保留较大字符；3) 输出。
- 思路2：1) 输入的字符进行合法性检查，并与（某寄存器中的）最大字符进行比较，保留两者的最大值，然后存储输入字符；2) 输出输入的字符串和最大值。

问题：哪一种方法更快？

(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）然后转换成十六进制数，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>

按任一键重做；按 Esc 键返回主菜单。

```
please input the function number (1~5): 3
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" [ESC/any other key]:
```

关键点

- 输入：多字符处理、非法判断（超过255、非数字字符）、**ASCII与数字转换**
- 处理：排序、**例4.43**（2层循环、冒泡法）
- 输出：各位数值10进制及16进制显示，**例4.49、例4.51**

(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。

按任一键重新对时；按 Esc 键返回主菜单。

说明：只完成显示功能即可，位置不做要求。

```
press anykey to display the time
16: 11: 03
What do you want to do next" Main Menu or Redo" [ESC/any other key]:
```

提示

- 利用DOS中断INT 21H的2CH号功能

作业说明

- 提交内容
 - 代码文件(**xxx.asm**)
 - 实验报告(**xxx.pdf**)
 - 实验环境（软件、硬件）、问题分析、解决方法（包含流程图）、代码实现、遇到的问题及调试
- 命名规则：学号_姓名，如**1111_张三.asm**
- 统一在学在西电（西电**SPOC**）线上提交

作业说明

- 独立完成
- 可以讨论、交流
- **不允许抄袭、盗用别人代码**
- 抄袭后果：抄袭者和被抄袭者均判0分

Q & A