
微机原理与接口技术大作业



学生姓名： 吴程锴

学 号： 18029100040

班 级： 1802015

授课教师： 张 葵

提交日期： 2020 年 12 月 20 日

目录

一、 实验要求.....	1
二、 程序设计.....	1
2.1 任务 1：将字符串中的小写字母变换成大写字母.....	3
2.1.1 主框架.....	3
2.1.2 输入字符串函数.....	3
2.2 任务 2：在字符中找最大值.....	4
2.2.1 主框架.....	4
2.3 任务 3：输入数据组的排序.....	5
2.3.1 主框架.....	5
2.3.2 字符串转化为数组函数.....	6
2.3.3 十六进制显示十进制数函数.....	7
2.3.4 显示整个数组函数.....	8
2.3.5 数组排序函数.....	9
2.4 任务 4：时间实时显示.....	10
2.4.1 主框架.....	10
2.4.2 时间校准函数.....	11
2.4.3 十进制显示函数.....	11
2.4.4 时间显示函数.....	12
2.4.5 实时显示时间函数.....	13
2.5 任务 5：返回操作系统.....	14
2.5.1 主框架.....	14
2.6 设置系统页面及整合功能.....	15
2.6.1 主框架.....	15
2.6.2 显示主菜单.....	15
三、 功能测试.....	16
3.1 将字符串中的小写字母变换成大写字母.....	17
3.2 在字符中找最大值.....	18
3.3 输入数据组的排序.....	19
3.4 时间的显示.....	20
3.5 返回操作系统.....	21

四、 总结.....	22
附录.....	23

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。

◆ 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

◆ 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

◆ 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

◆ 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。

按任意键重做；按 Esc 键返回主菜单

◆ 按数字键“5”，结束程序的运行，返回操作系统

二、程序设计

采用分模块的方式，把前四个任务分别写成四个函数，最后设计系统页面，整合所有功能。接下来分别说明四个任务和系统页面的设计。

数据段和堆栈段定义如下所示

```
1. DATAS SEGMENT
2.     TIPMenu1 DB 'Push 1 To Capitalize String','$'
3.     TIPMenu2 DB 'Push 2 To Find The Maximum of The String','$'
4.     TIPMenu3 DB 'Push 3 To Sort The Array','$'
5.     TIPMenu4 DB 'Push 4 To Show Time in Real Time','$'
```

```

6.      TIPMenu5 DB 'Push 5 To Exit','$'
7.
8.      TIPInputString DB 'Please input string:','$'
9.      TIPOriginalString DB 'Original string:','$'
10.     TIPNewString DB 'New string:','$'
11.
12.     TIPMax DB 'The maximum is ','$'
13.
14.     TIPInputArray DB 'Please input array:','$'
15.     TIPOriginalArray DB 'Original array:','$'
16.     TIPNewArray DB 'New array:','$'
17.
18.     TIPInputTIME DB 'Please input TIME(HH:MM:SS):','$'
19.
20.
21.     BUFFSTRING DB 100
22.                DB ?
23.                DB 100 DUP(?)
24.     ARRAY DB 100 DUP(0)
25.     ARRAYLength DB 0
26.     ARRAYSorted DB 100 DUP(0)
27.     SpiltSymbol DB ?
28.     TIMEBUFF DB 3 DUP(0)
29.
30.     TEMP DB 0
31.     Decimal DB 10
32.     HEX DB 16
33.     ;此处输入数据段代码
34. DATAS ENDS
35.
36. STACKS SEGMENT
37.     DB 100 DUP(?)
38.     TOP LABEL WORD
39.     ;此处输入堆栈段代码
40. STACKS ENDS

```

在数据段中，TIPxx 用于显示的提示信息或者结果，BUFFSTRING 用于存放输入进来的字符串，ARRAY 用于存放从字符串转码来的十进制数组，ARRAYLength 用于存放数组的长度，ARRAYSorted 用于存放按增序排好的数组，SpiltSymbol 用于存放用于隔开数据的字符（在本文中“，”用于隔开数组中的数据，“:”用于隔开时间），TEMP 用于临时存放数据，Decimal 用于输出十进制数时的除数，HEX 用于输出十六进制数时的除数。

在堆栈段中，开辟 100 个字节的空間，用 TOP 可以采用字型访问该空間。

2.1 任务 1：将字符串中的小写字母变换成大写字母

2.1.1 主框架

程序主要使用遍历访问的方法，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。该任务模块的基本思路如图 1 所示。最后把程序封装为函数 *CapitalizeString*，无入口和出口参数，在函数的开头和结尾加上保护现场和恢复现场。该任务模块的基本思路如图 1 所示

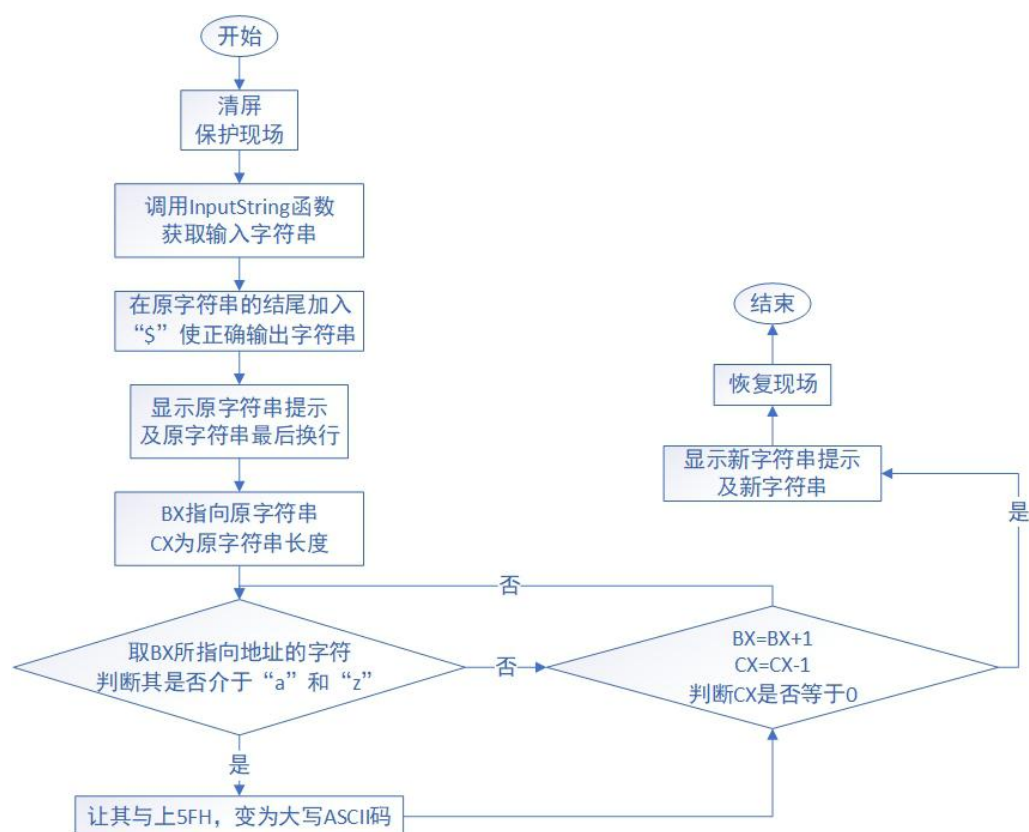


图 1 任务一主流程图

2.1.2 输入字符串函数

入口参数：SI 指向输入提示语句

出口参数：无

流程图中的 *InputString* 函数使用 DOS 系统功能调用中的字符串显示和输入和字符显示。字符串显示用于显示 *TIPInputString*，在输入完字符串后使用字符显示换行。

InputString 流程图如图 2 所示



图 2 *InputString* 函数流程图

2.2 任务 2：在字符串中找最大值

2.2.1 主框架

程序主要使用遍历访问的方法，如果当前字符的 ASCII 码大于所记录的最大值，则将此字符的 ASCII 码记录下来，最终得到字符串中的最大值，最后把程序封装为函数 *FindMaxStr*，无入口和出口参数，在函数的开头和结尾加上保护现场和恢复现场。该任务模块的基本思路如图 3 所示

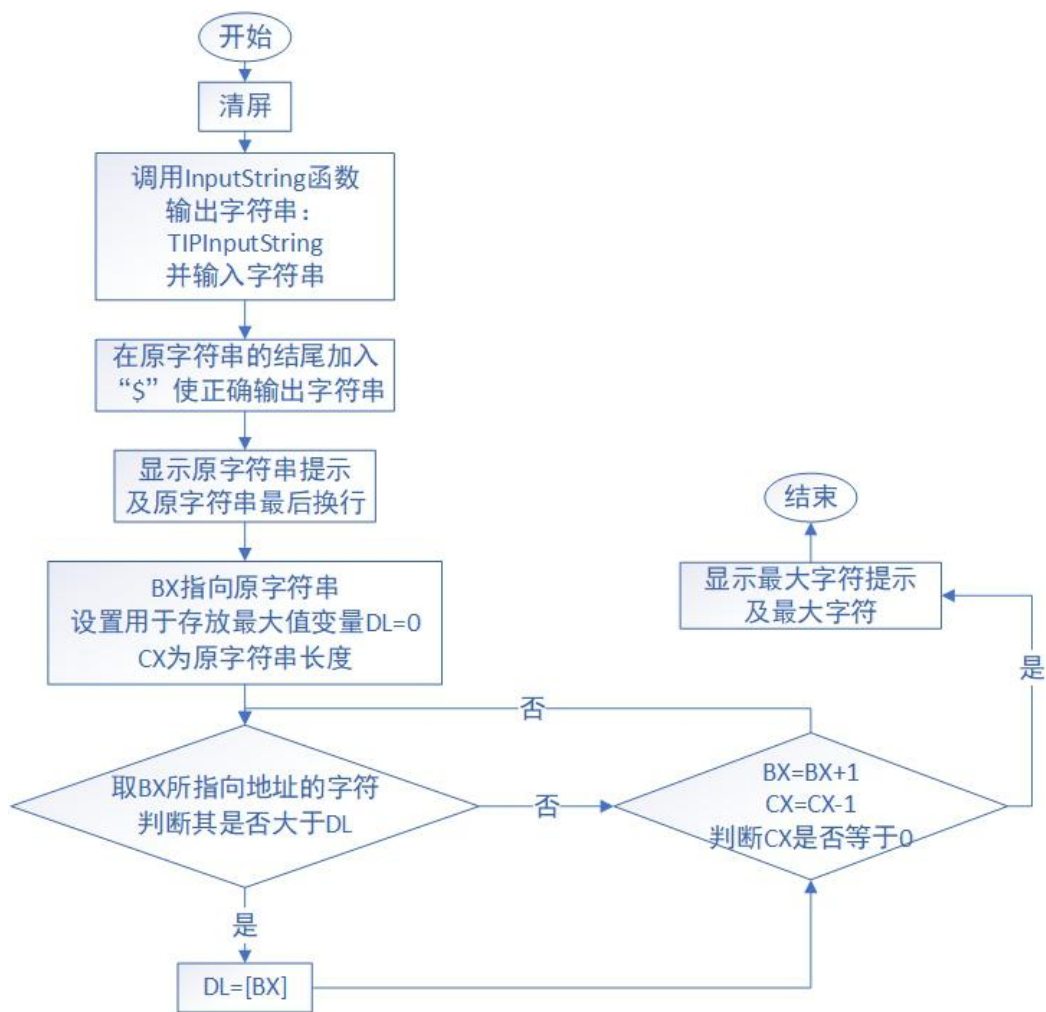


图 3 任务 2 主流程图

2.3 任务 3：输入数据组的排序

2.3.1 主框架

程序的数组输入调用 *InputString* 函数，数据以逗号分隔，字符串数组转换函数 *Str2Num* 把输入的字符串以 “,” 为分隔，并将对应的十进制数存入 *ARRAY* 中；十进制转十六进制输出部分由 *Disp* 函数完成；整个数组的显示由 *DISPARRAY* 函数完成；数组排序由 *SortARRAY* 函数完成，排好序的数组存入 *ARRAYSorted* 中。最后把程序封装为函数 *FindMaxStr*，无入口和出口参数，在函数的开头和结尾加上保护现场和恢复现场。该任务模块的基本思路如图 4 所示



图 4 任务 3 主流程图

2.3.2 字符串转化为数组函数

入口参数：分隔符

出口参数：无

函数主要用遍历的方式查找分隔符“，”，当找到分隔符就开始处理分隔符前的数据，从高位开始，加上当前位，移到下一位，乘 10 后加上该位，直到该数据结束。遍历整个字符串，把所有数据存入 ARRAY 中。Str2Num 函数流程图如图 5 所示

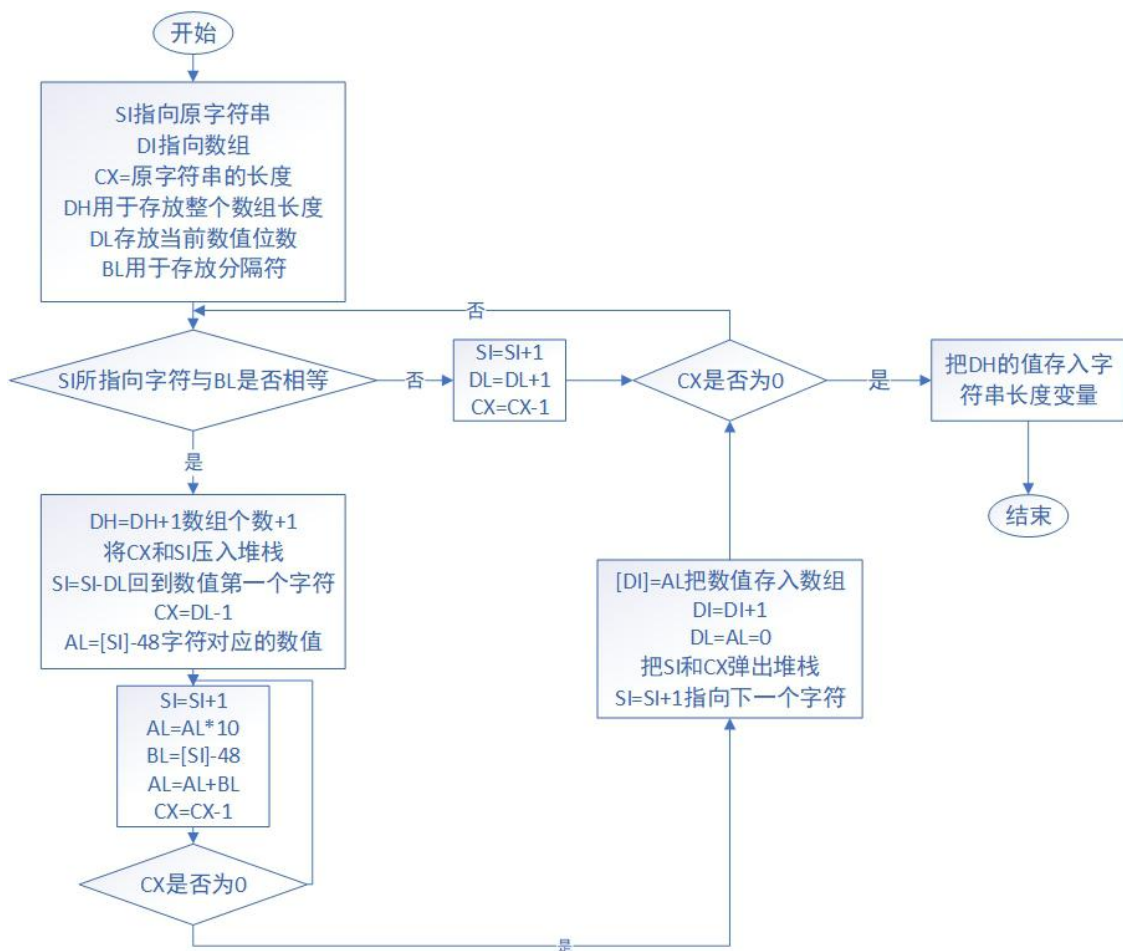


图 5 Str2Num 函数流程图

2.3.3 十六进制显示十进制数函数

入口参数：SI 指向十进制数地址

出口参数：无

函数的方法是十进制数对十六取余，并将余数压入堆栈，再对商取余，将余数压入堆栈，直到商为 0，最后把余数一个一个弹出，判断是否小于等于 9，如果小于等于 9，则加上 48 得到对应的 ASCII 码输出，如果大于 9，则加上 65 减去 10 得到对应的 ASCII 码输出。Disp 函数流程图如图 6 所示

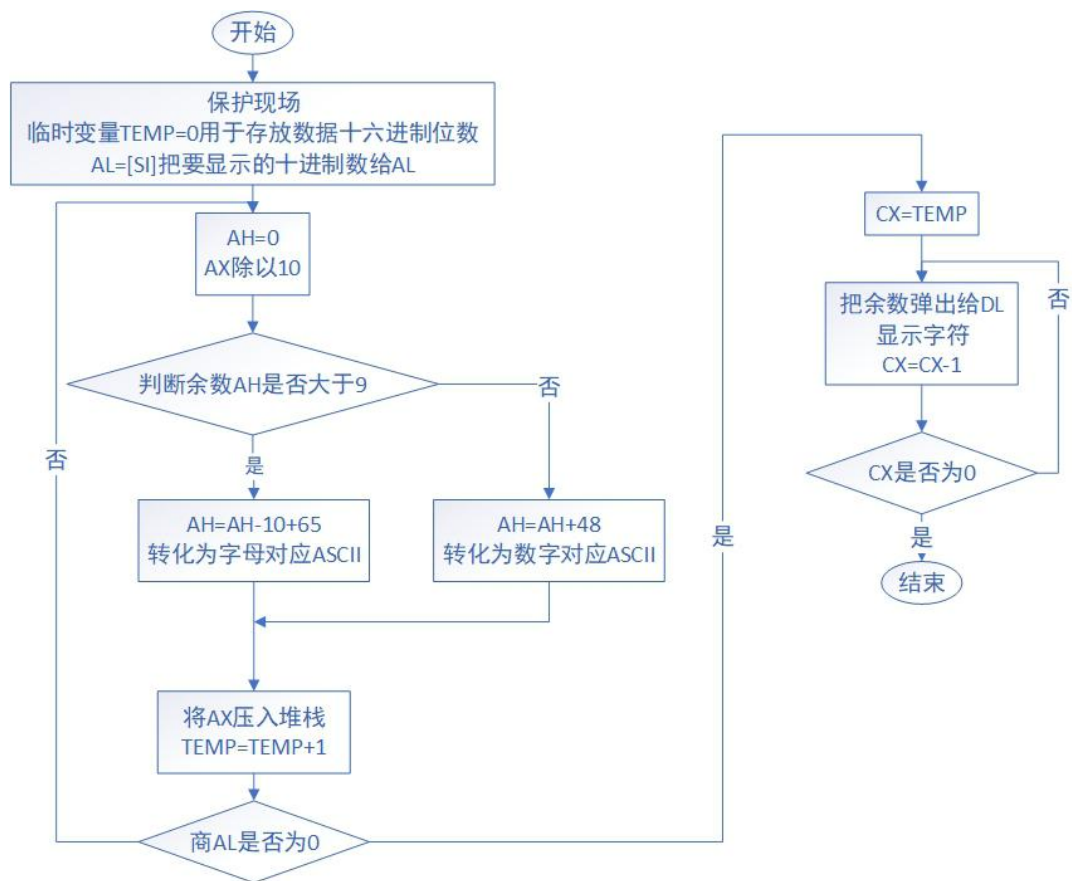


图 6 *Disp* 函数流程图

2.3.4 显示整个数组函数

入口参数：SI 指向数组首地址

出口参数：无

函数主要遍历数组，并调用 *Disp* 函数以十六进制显示，并在数据之间用逗号分隔。

DISPARRAY 函数流程图如图 7 所示

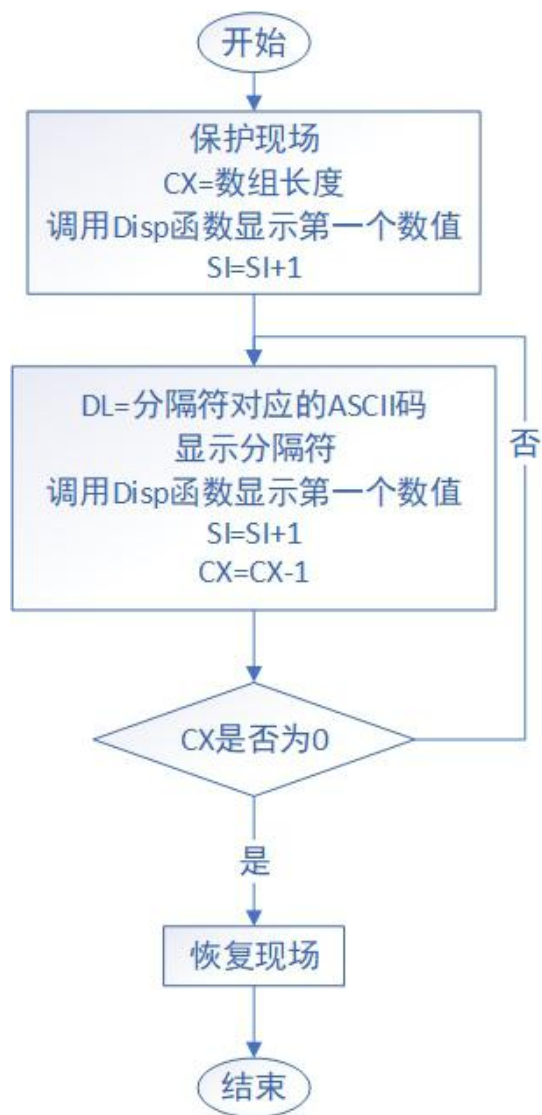


图 7 *DISPARRAY* 函数流程图

2.3.5 数组排序函数

入口参数：无

出口参数：无

函数主要使用遍历的方式，每次遍历找到数组中最小的数据，放入新数组中，并把原数组中的数据变为 255，一共遍历数组长度此，最终得到从小到大排列的新数组。

SortARRAY 函数流程图如图 8 所示

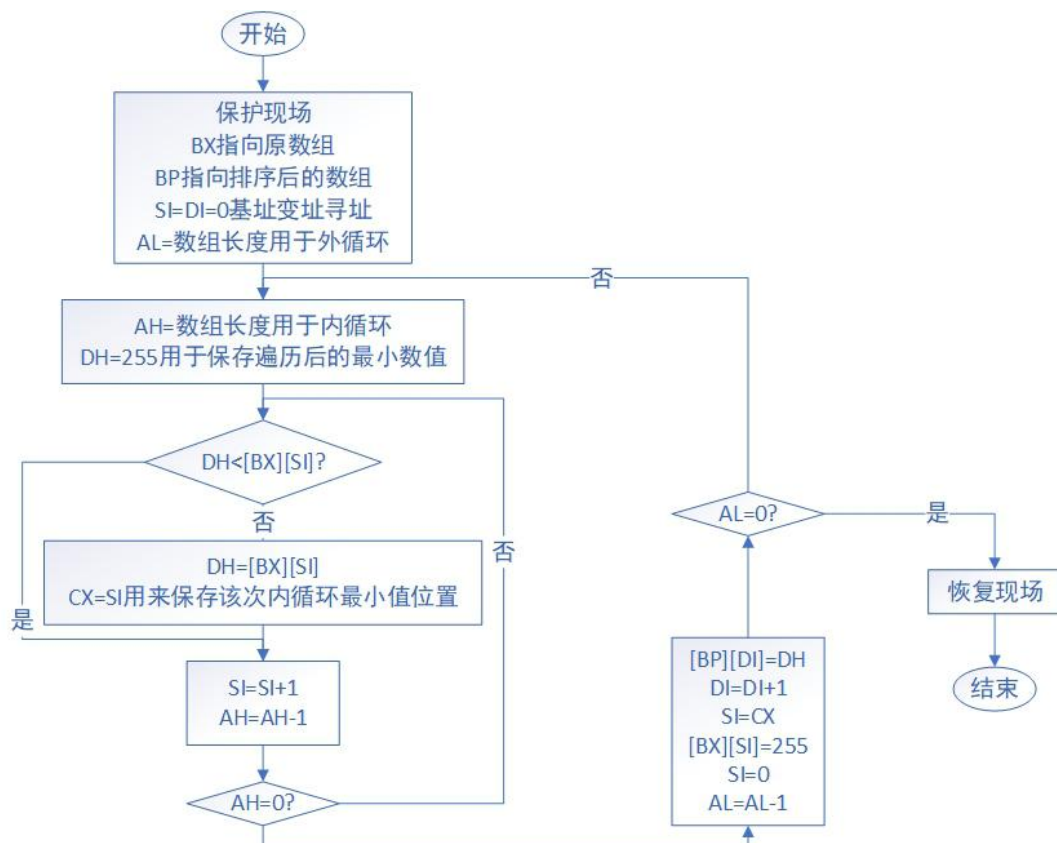


图 8 SortARRAY 函数流程图

2.4 任务 4：时间实时显示

2.4.1 主框架

首先调用 *CorrectTIME* 函数进行时间校准，函数中调用了 *InputString* 函数，数据以冒号分隔，字符串数组转换函数 *Str2Num* 把输入的字符串以 “:” 为分隔，并将对应的十进制数存入 *ARRAY* 中，使用置系统时间功能重置系统时间；之后使用 *DispRealTIME* 函数实时显示时间，时间的显示调用了 *DispTIME* 函数，其中的十进制显示调用了 *DispDec* 函数。最后把程序封装为函数 *ShowTime*，无入口和出口参数，在函数的开头和结尾加上保护现场和恢复现场。该任务模块的基本思路如图 9 所示

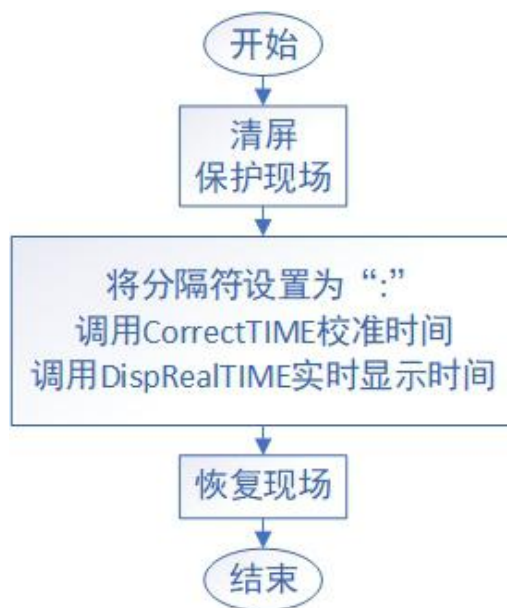


图 9 任务 4 主流程图

2.4.2 时间校准函数

时间校准函数首先显示校准时间提示，接着调用了 *InputString* 来获取输入的时间信息，再调用 *Str2Num* 函数把字符串转化为数据信息保存，最后调用置系统时间功能进行时间校准。*CorrectTIME* 函数流程图如图 10 所示

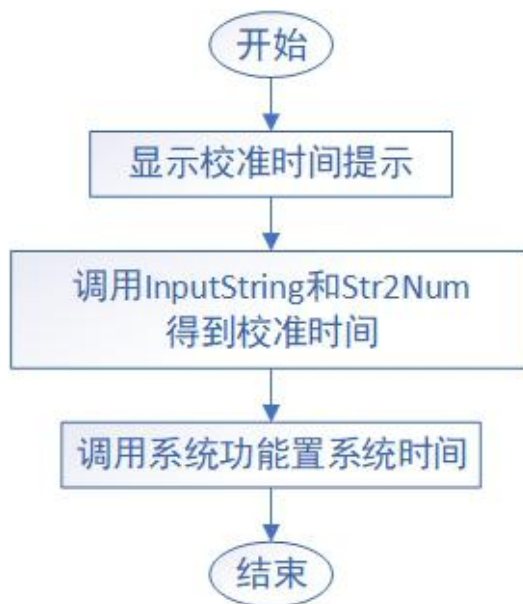


图 10 *CorrectTIME* 函数流程图

2.4.3 十进制显示函数

入口参数：SI 指向时间数据地址

出口参数：无

由于时间由两位构成，所以不需要像 *Disp* 函数那样考虑显示多位的情况，只需要对数据进行除 10 运算后得到的余数和商加上 48 即可得到对应的 ASCII 码，最后显示两个字符即可。*DispDec* 函数流程图如图 11 所示



图 11 *DispDec* 函数流程图

2.4.4 时间显示函数

入口参数：SI 指向时间数据地址

出口参数：无

主要调用 *DispDec* 函数，并在两数据间显示“:”分隔。*DispTIME* 函数流程图如图 12 所示



图 12 *DispTIME* 函数流程图

2.4.5 实时显示时间函数

DispRealTIME 函数首先设置光标位置到右上角, 接着调用 DOS 系统功能获取系统时间, 将获取的时间存入 ARRAY, 调用 *DispTIME* 函数显示时间, 接着给 CX 赋值为 0FFFFH 进行延时操作, 最后调用键盘服务读取键盘状态, 若有输入则停止, 无输入则更新显示时间。 *DispRealTIME* 函数流程图如图 13 所示

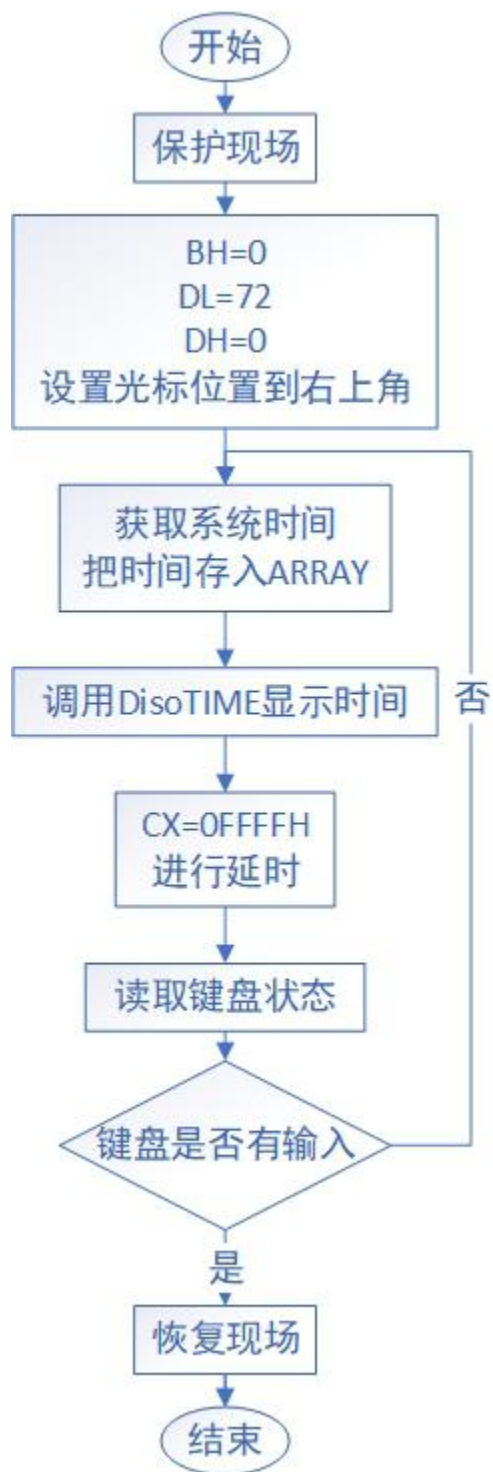


图 13 *DispRealTIME* 函数流程图

2.5 任务 5：返回操作系统

2.5.1 主框架

清屏后调用 DOS 功能中的终止进程功能。

2.6 设置系统页面及整合功能

2.6.1 主框架

调用 *Menu* 函数显示主菜单，设置光标到左上角，获取键盘状态，比较输入的字符分别与 1、2、3、4、5 进行对比跳转到对应任务，如果不是 1、2、3、4、5 则重新获取键盘状态；跳转到对应任务后调用相应任务函数，接着检测键盘输入是否为 Esc，如果是则返回程序开始，显示主菜单，否则返回对应任务重新执行。该模块的基本思路如图 14 所示

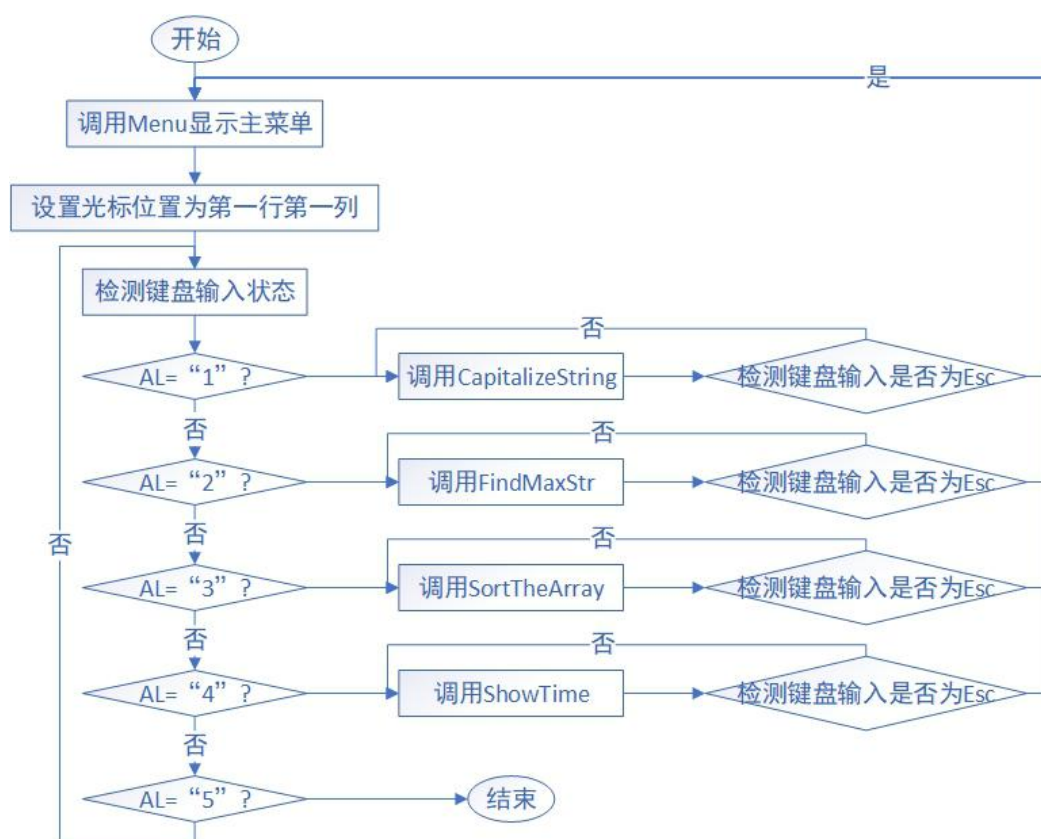


图 14 整体框架主流程图

2.6.2 显示主菜单

首先清屏，设置光标位置为第二行第一列，开始调用字符串输出功能，输出 5 个 TIP。 *Menu* 函数流程图如图 15 所示

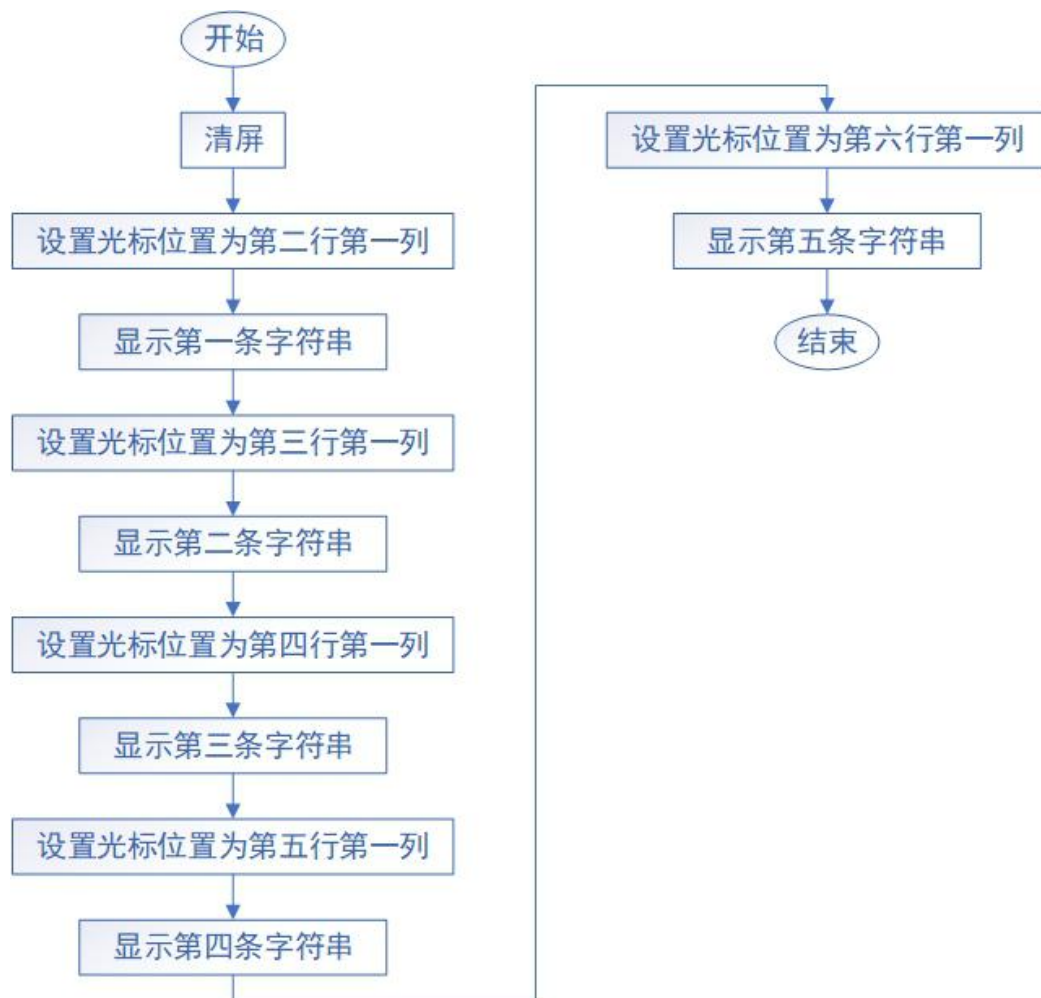


图 15 Menu 函数流程图

三、功能测试

到此为止，所有功能及整体框架全部完成，完整代码见附录。

运行后主菜单页面如所示，输入 1~5 完成不同功能。

```
Push 1 To Capitalize String
Push 2 To Find The Maximum of The String
Push 3 To Sort The Array
Push 4 To Show Time in Real Time
Push 5 To Exit
```

图 16 主菜单页面

3.1 将字符串中的小写字母变换成大写字母

按数字键 1 进入任务 1，任务 1 输入页面如图 17 所示

```
Please input string: _
```

图 17 任务 1 输入页面

输入“abCD12#%”后按回车，结果如图 18 所示

```
Original string:abCD12#%  
New string:ABCD12#%
```

图 18 任务 1 结果

按 Esc 键返回主菜单

3.2 在字符中找最大值

按数字键 2 进入任务 2，任务 2 输入页面如所示

```
Please input string: _
```

图 19 任务 2 输入页面

输入“123abcABC”后按回车，结果如图 20 所示

```
Original string:123abcABC  
The maximum is c_
```

图 20 任务 2 结果

按 Esc 键返回主菜单

3.3 输入数据组的排序

按数字键 3 进入任务 3，任务 3 输入页面如所示

```
Please input array:
```

图 21 任务 3 输入页面

输入 “12,13,14,11,10,” 后按回车，结果如图 22 所示

```
Original array:C,D,E,B,A  
New string:A,B,C,D,E
```

图 22 任务 3 结果

按 Esc 键返回主菜单

3.4 时间的显示

按数字键 4 进入任务 4，任务 4 输入页面如所示

```
Please input TIME(HH:MM:SS):
```

图 23 任务 4 输入页面

输入“14:43:30:”后按回车，结果如图 24 和图 25 所示

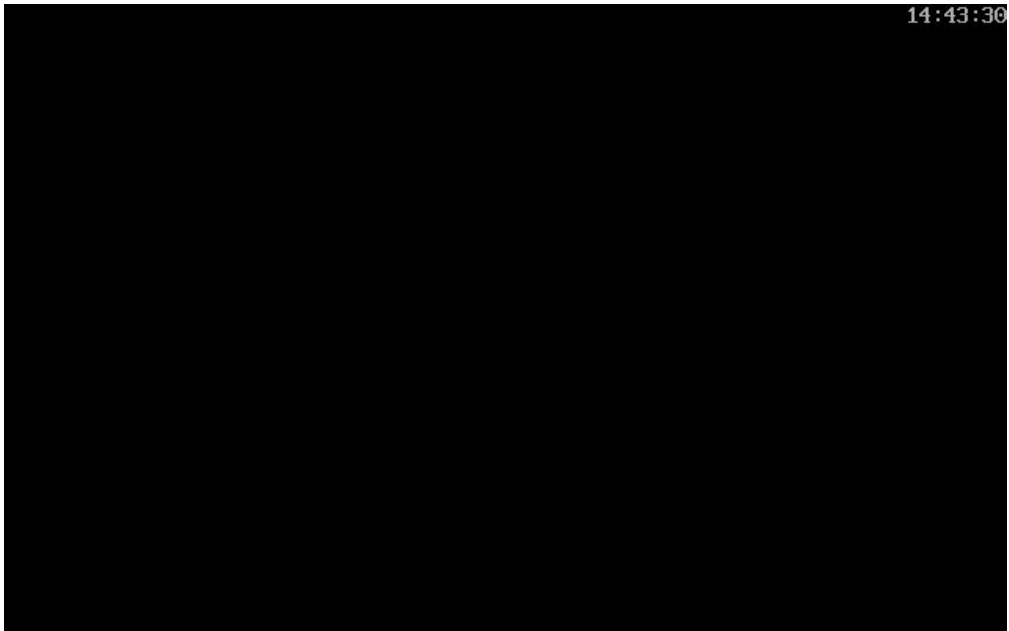


图 24 任务 4 结果 1

过一段时间后

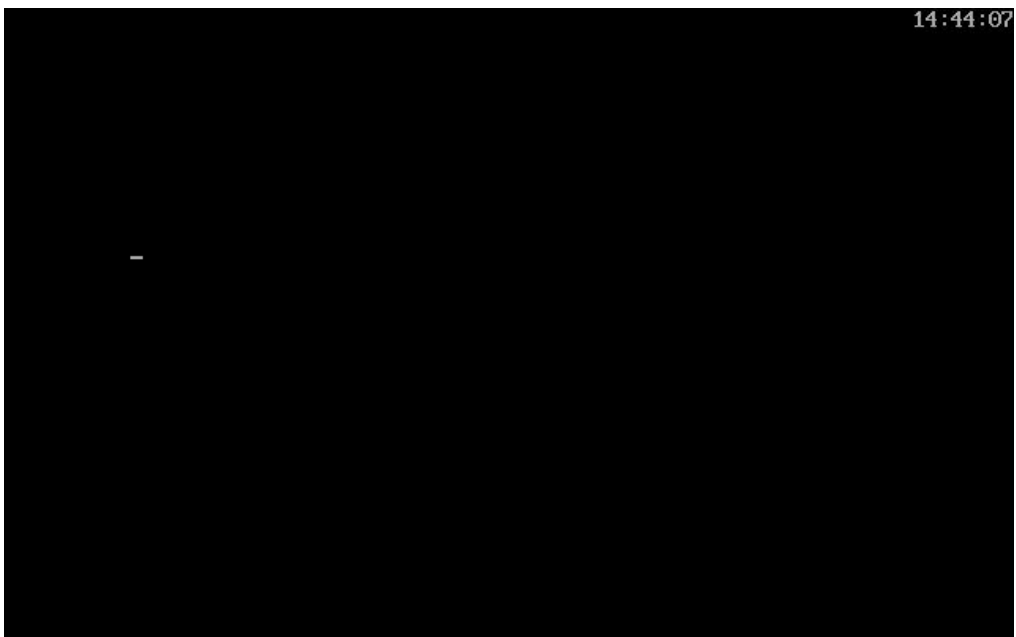


图 25 任务 4 结果 2

按 Esc 键返回主菜单

3.5 返回操作系统

按数字键 5 进入任务 5，返回操作系统，结果如图 26 所示

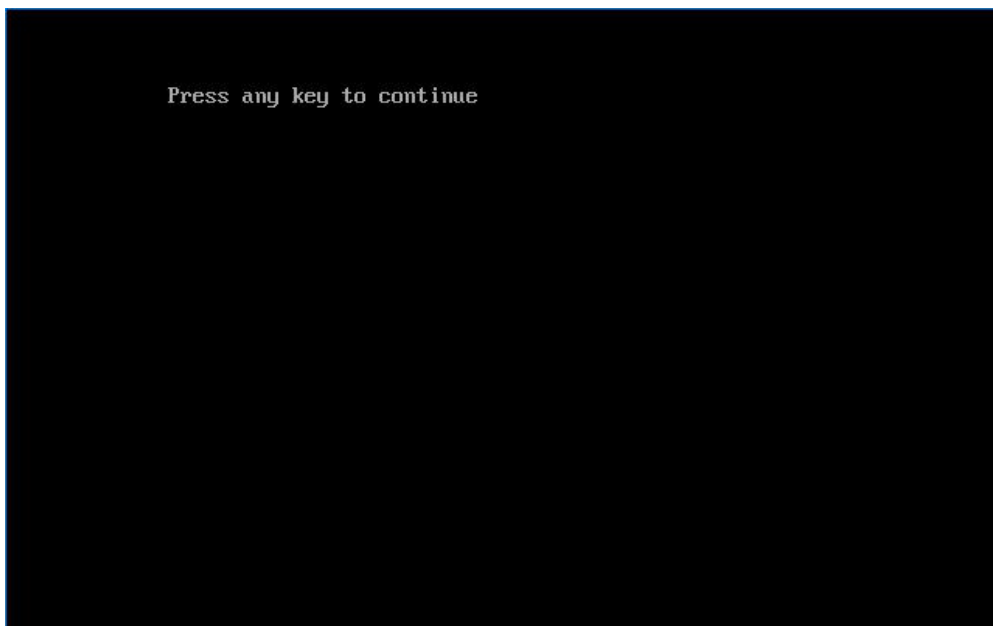


图 26 任务 5 结果

四、总结

所有模块均按要求正常运行无任何 BUG。

通过这次的大作业，我对汇编中寻址方式、运算操作、位操作、分支结果、循环结构、子程序设计等有了更加熟悉的认识。

附录

```
1. DATAS SEGMENT
2.     TIPMenu1 DB 'Push 1 To Capitalize String','$'
3.     TIPMenu2 DB 'Push 2 To Find The Maximum of The String','$'
4.     TIPMenu3 DB 'Push 3 To Sort The Array','$'
5.     TIPMenu4 DB 'Push 4 To Show Time in Real Time','$'
6.     TIPMenu5 DB 'Push 5 To Exit','$'
7.
8.     TIPInputString DB 'Please input string:','$'
9.     TIPOriginalString DB 'Original string:','$'
10.    TIPNewString DB 'New string:','$'
11.
12.    TIPMax DB 'The maximum is ','$'
13.
14.    TIPInputArray DB 'Please input array:','$'
15.    TIPOriginalArray DB 'Original array:','$'
16.    TIPNewArray DB 'New array:','$'
17.
18.    TIPInputTIME DB 'Please input TIME(HH:MM:SS):','$'
19.
20.
21.    BUFFSTRING DB 100
22.                DB ?
23.                DB 100 DUP(?)
24.    ARRAY DB 100 DUP(0)
25.    ARRAYLength DB 0
26.    ARRAYSorted DB 100 DUP(0)
27.    SpiltSymbol DB ?
28.    TIMEBUFF DB 3 DUP(0)
29.
30.    TEMP DB 0
31.    Decimal DB 10
32.    HEX DB 16
33.    ;此处输入数据段代码
34. DATAS ENDS
35.
36. STACKS SEGMENT
37.     DB 100 DUP(?)
38.     TOP LABEL WORD
39.     ;此处输入堆栈段代码
40. STACKS ENDS
41.
```

```

42. CODES SEGMENT
43.     ASSUME CS:CODES,DS:DATAS,SS:STACKS
44. START:
45.     MOV AX,DATAS
46.     MOV DS,AX
47.
48. ReturnMenu:
49.     CALL Menu
50.
51. Redo:
52.     MOV AH,2
53.     MOV BH,0
54.     MOV DL,0
55.     MOV DH,0
56.     INT 10H
57.
58.     MOV AH,1
59.     INT 21H
60.
61.
62.     CMP AL,'1'
63.     JZ MISSION1
64.     CMP AL,'2'
65.     JZ MISSION2
66.     CMP AL,'3'
67.     JZ MISSION3
68.     CMP AL,'4'
69.     JZ MISSION4
70.     CMP AL,'5'
71.     JZ MISSION5
72.     JMP Redo
73.
74. MISSION1:
75.     CALL CapitalizeString
76.     ;检测键盘输入是否为 Esc, 若是则返回主菜单, 否则重做
77.     MOV AH,8
78.     INT 21H
79.     CMP AL,1BH
80.     JZ ReturnMenu
81.     JMP MISSION1
82.
83. MISSION2:
84.     CALL FindMaxStr
85.     ;检测键盘输入是否为 Esc, 若是则返回主菜单, 否则重做
86.     MOV AH,8

```

```

87.    INT 21H
88.    CMP AL,1BH
89.    JZ ReturnMenu
90.    JMP MISSION2
91.
92. MISSION3:
93.    CALL SortTheArray
94.    ;检测键盘输入是否为 Esc，若是则返回主菜单，否则重做
95.    MOV AH,8
96.    INT 21H
97.    CMP AL,1BH
98.    JZ ReturnMenu
99.    JMP MISSION3
100.
101. MISSION4:
102.    CALL ShowTime
103.    ;检测键盘输入是否为 Esc，若是则返回主菜单，否则重做
104.    MOV AH,8
105.    INT 21H
106.    CMP AL,1BH
107.    JZ ReturnMenu
108.    JMP MISSION4
109.
110. MISSION5:
111.    ;此处输入代码段代码
112.    ;清屏
113.    MOV AH,0
114.    MOV AL,3
115.    MOV BL,0
116.    INT 10H
117.    MOV AH,4CH
118.    INT 21H
119.
120. ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;函数定义
    区;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
121. Menu PROC FAR
122.    ;清屏
123.    MOV AH,0
124.    MOV AL,3
125.    MOV BL,0
126.    INT 10H
127.    ;第一条提示
128.    MOV AH,2
129.    MOV BH,0
130.    MOV DL,0

```

```

131.    MOV DH,1
132.    INT 10H
133.    MOV AH,9
134.    LEA DX,TIPMenu1
135.    INT 21H
136.    ;第二条提示
137.    MOV AH,2
138.    MOV BH,0
139.    MOV DL,0
140.    MOV DH,2
141.    INT 10H
142.    MOV AH,9
143.    LEA DX,TIPMenu2
144.    INT 21H
145.    ;第三条提示
146.    MOV AH,2
147.    MOV BH,0
148.    MOV DL,0
149.    MOV DH,3
150.    INT 10H
151.    MOV AH,9
152.    LEA DX,TIPMenu3
153.    INT 21H
154.    ;第四条提示
155.    MOV AH,2
156.    MOV BH,0
157.    MOV DL,0
158.    MOV DH,4
159.    INT 10H
160.    MOV AH,9
161.    LEA DX,TIPMenu4
162.    INT 21H
163.    ;第五条提示
164.    MOV AH,2
165.    MOV BH,0
166.    MOV DL,0
167.    MOV DH,5
168.    INT 10H
169.    MOV AH,9
170.    LEA DX,TIPMenu5
171.    INT 21H
172.
173.    RET
174. Menu ENDP
175. ;把十进制数输出

```

```
176. ;入口参数: 无
177. ;出口参数: 无
178. ShowTime PROC FAR
179.     PUSH AX
180.     PUSH BX
181.     PUSH CX
182.     PUSH DX
183.     PUSH SP
184.     PUSH BP
185.     PUSH SI
186.     PUSH DI
187.
188.     ;清屏
189.     MOV AH,0
190.     MOV AL,3
191.     MOV BL,0
192.     INT 10H
193.
194.     MOV AH,2
195.     MOV BH,1
196.     MOV DL,0
197.     MOV DH,0
198.     INT 10H
199.
200.     MOV [SpiltSymbol],':'
201.     CALL CorrectTIME
202.     CALL DispRealTIME
203.
204.     POP DI
205.     POP SI
206.     POP BP
207.     POP SP
208.     POP DX
209.     POP CX
210.     POP BX
211.     POP AX
212.     RET
213. ShowTime ENDP
214. ;把十进制数输出
215. ;入口参数: 十进制数地址 SI
216. ;出口参数: 无
217. DispDec PROC FAR
218.     PUSH AX
219.     PUSH BX
220.     PUSH CX
```

```

221.    PUSH DX
222.
223.    MOV AL,[SI]
224.    CBW
225.    DIV Decimal
226.    ADD AH,'0'
227.    MOV BH,AH
228.    ADD AL,'0'
229.
230.    MOV DL,AL
231.    MOV AH,02H
232.    INT 21H
233.
234.    MOV DL,BH
235.    INT 21H
236.
237.    POP DX
238.    POP CX
239.    POP BX
240.    POP AX
241.    RET
242. DispDec ENDP
243. ;输出时间
244. ;入口参数: 时间缓冲区 ARRAY
245. ;出口参数: 无
246. DispTIME PROC FAR
247.    PUSH SI
248.    LEA SI,ARRAY
249.    CALL DispDec
250.
251.    MOV DL,':'
252.    MOV AH,02H
253.    INT 21H
254.
255.    INC SI
256.    CALL DispDec
257.
258.    MOV DL,':'
259.    MOV AH,02H
260.    INT 21H
261.
262.    INC SI
263.    CALL DispDec
264.    POP SI
265.    RET

```

```

266. DISPTIME ENDP
267. ;时间矫正时间
268. ;入口参数: 无
269. ;出口参数: 无
270.     CorrectTIME PROC FAR
271.         LEA DX,TIPInputTIME
272.         MOV AH,9
273.         INT 21H
274.         CALL InputString
275.         CALL Str2Num
276.
277.         LEA SI,ARRAY
278.         MOV CH,[SI]
279.         INC SI
280.         MOV CL,[SI]
281.         INC SI
282.         MOV DH,[SI]
283.         MOV DL,0
284.         MOV AH,2DH
285.         INT 21H
286.         RET
287.     CorrectTIME ENDP
288. ;实时输出时间
289. ;入口参数: 时间缓冲区 ARRAY
290. ;出口参数: 无
291. DispRealTIME PROC FAR
292.     PUSH SI
293.     ;清屏
294.     MOV AH,0
295.     MOV AL,3
296.     MOV BL,0
297.     INT 10H
298.     DISPREALTIME1:
299.         MOV AH,2
300.         MOV BH,0
301.         MOV DL,72
302.         MOV DH,0
303.         INT 10H
304.
305.         MOV AH,2CH
306.         INT 21H
307.         MOV [ARRAY],CH
308.         MOV [ARRAY+1],CL
309.         MOV [ARRAY+2],DH
310.

```



```

311.          CALL DispTIME
312.
313.          MOV AH,2
314.          MOV DX,090AH
315.          MOV BH,0
316.          INT 10H
317.
318.          MOV CX,0FFFFH
319.          DELAY:LOOP DELAY
320.          MOV AH,01H
321.          INT 16H
322.          JNZ StopShow
323.          JMP DISPREALTIME1
324.  StopShow:
325.
326.          POP SI
327.          RET
328. DispRealTIME ENDP
329. ;把数组按升序排列，十六进制输出
330. ;入口参数: 无
331. ;出口参数: 无
332. SortTheArray PROC FAR
333.          PUSH AX
334.          PUSH BX
335.          PUSH CX
336.          PUSH DX
337.          PUSH SP
338.          PUSH BP
339.          PUSH SI
340.          PUSH DI
341.
342.          ;清屏
343.          MOV AH,0
344.          MOV AL,3
345.          MOV BL,0
346.          INT 10H
347.
348.          MOV AH,2
349.          MOV BH,1
350.          MOV DL,0
351.          MOV DH,0
352.          INT 10H
353.
354.          MOV [SpiltSymbol],',';输入以逗号分隔
355.          ;输入字符串

```

```

356.    CALL InputString
357.    ;清屏
358.    MOV AH,0
359.    MOV AL,3
360.    MOV BL,0
361.    INT 10H
362.    ;把字符串转为数字放入数组
363.    CALL Str2Num
364.    ;输出原数组
365.    LEA DX,TIPOriginalArray
366.    MOV AH,9
367.    INT 21H
368.
369.    MOV AL,[ARRAYLength]
370.    MOV AH,0
371.    MOV CX,AX
372.    LEA SI,ARRAY
373.    CALL DISPARRAY
374.
375.    MOV DL,10
376.    MOV AH,02H
377.    INT 21H
378.    ;递增排序
379.    CALL SortARRAY
380.
381.
382.    ;输出新数组
383.    LEA DX,TIPNewString
384.    MOV AH,9
385.    INT 21H
386.
387.    LEA SI,ARRAYSorted
388.    CALL DISPARRAY
389.
390.    POP DI
391.    POP SI
392.    POP BP
393.    POP SP
394.    POP DX
395.    POP CX
396.    POP BX
397.    POP AX
398.    RET
399. SortTheArray ENDP
400. ;把字符串转为数字放入数组,以逗号分隔和结尾

```

```

401. ;入口参数: 无
402. ;出口参数: 无
403. Str2Num PROC FAR
404.     MOV AL,BUFFSTRING+1
405.     CBW
406.     MOV CX,AX
407.
408.     LEA SI,BUFFSTRING+2
409.     LEA DI,ARRAY
410.     MOV DH,0           ;用于存放数组个数
411.     MOV DL,0           ;用于存放该数字位数
412.     MOV AL,0           ;用于存放该数字十进制
413.     Spilt:
414.         MOV BL,[SpiltSymbol]
415.         CMP BYTE PTR [SI],BL
416.         JZ AddNum
417.         INC SI
418.         INC DL
419.         JMP NotSpilt
420.     AddNum:
421.         INC DH
422.         PUSH CX
423.         PUSH SI
424.
425.         MOV AL,DL
426.         CBW
427.         MOV CX,AX
428.
429.     Return:
430.         DEC SI
431.         ;MOV DL,[SI]
432.         ;MOV AH,02H
433.         ;INT 21H
434.         LOOP Return
435.
436.     MOV CX,AX
437.     DEC CX
438.
439.     MOV AL,0
440.     MOV AH,0
441.
442.     MOV AL,[SI]
443.     SUB AL,'0'
444.
445.

```

```

446.
447.      Str2Dec:
448.          INC SI
449.          MUL Decimal
450.          MOV BL,[SI]
451.          SUB BL,'0'
452.          ADD AL,BL
453.          LOOP Str2Dec
454.
455.          MOV [DI],AL
456.          INC DI
457.          MOV DL,0
458.          MOV AL,0
459.          POP SI
460.          INC SI;指向逗号下一个
461.          POP CX
462.          NotSpilt:
463.          LOOP Spilt
464.          MOV [ARRAYLength],DH
465.          RET
466. Str2Num ENDP
467. ;递增排序
468. ;入口参数: 无
469. ;出口参数: 无
470. SortARRAY PROC FAR
471.     LEA BX,ARRAY
472.     LEA BP,ARRAYSorted
473.     MOV SI,0
474.     MOV DI,0
475.
476.     MOV AL,[ARRAYLength];外循环
477.     SORT:
478.         MOV AH,[ARRAYLength]
479.         MOV DH,127
480.         FindMin:
481.             CMP DH,[BX][SI]
482.             JB FindMinNEXT
483.             MOV DH,[BX][SI]
484.             MOV CX,SI
485.             FindMinNEXT:
486.                 INC SI
487.                 DEC AH
488.                 JNZ FindMin
489.             MOV [BP][DI],DH
490.             INC DI

```

```

491.      MOV SI,CX
492.      MOV DH,127
493.      MOV [BX][SI],DH
494.      MOV SI,0
495.      DEC AL
496.      JNZ SORT
497.
498.      ;MOV DL,[BX]
499.
500.      RET
501. SortARRAY ENDP
502. ;把十六进制数输出
503. ;入口参数: 十六进制数地址 SI
504. ;出口参数: 无
505. Disp PROC FAR
506.      PUSH CX
507.      MOV AL,[SI]
508.      NEXTNum:
509.          CBW
510.          DIV HEX
511.          CMP AH,9
512.          JA LETTER
513.          ADD AH,'0'
514.          JMP NotLETTER
515.      LETTER:
516.          SUB AH,10
517.          ADD AH,'A'
518.
519.      NotLETTER:
520.          PUSH AX
521.          INC [TEMP]
522.
523.          CMP AL,0
524.          JZ NoMore
525.
526.          JMP NEXTNum
527.
528.      NoMore:
529.          MOV AL,[TEMP]
530.          CBW
531.          MOV CX,AX
532.
533.      OUTNum:
534.          POP DX
535.          XCHG DL,DH

```

```

536.          MOV AH,02H
537.          INT 21H
538.          LOOP OUTNum
539.
540.
541.
542.    MOV [TEMP],0
543.    POP CX
544.    RET
545. Disp ENDP
546. ;输出数组
547. ;入口参数: 数组首地址 SI, 长度 CX
548. ;出口参数: 无
549. DISPARRRY PROC FAR
550.    PUSH AX
551.    PUSH BX
552.    PUSH CX
553.    PUSH DX
554.
555.    MOV AL,[ARRAYLength]
556.    MOV AH,0
557.    MOV CX,AX
558.    DEC CX
559.    CALL Disp
560.    INC SI
561.    DispNextNum:
562.        MOV DL,[SpiltSymbol]
563.        MOV AH,02H
564.        INT 21H
565.        CALL Disp
566.        INC SI
567.    LOOP DispNextNum
568.
569.    POP DX
570.    POP CX
571.    POP BX
572.    POP AX
573.    RET
574. DISPARRRY ENDP
575. ;把字符变为大写
576. ;入口参数: 无
577. ;出口参数: 无
578. FindMaxStr PROC FAR
579.    PUSH AX
580.    PUSH BX

```

```

581.    PUSH CX
582.    PUSH DX
583.    PUSH SP
584.    PUSH BP
585.    PUSH SI
586.    PUSH DI
587.
588.    ;清屏
589.    MOV AH,0
590.    MOV AL,3
591.    MOV BL,0
592.    INT 10H
593.
594.    MOV AH,2
595.    MOV BH,1
596.    MOV DL,0
597.    MOV DH,0
598.    INT 10H
599.    ;输入字符串
600.    CALL InputString
601.    ;清屏
602.    MOV AH,0
603.    MOV AL,3
604.    MOV BL,0
605.    INT 10H
606.    ;把输入字符串个数转化为字类型给 CX 来循环处理，并在结尾加上 '$'
607.    LEA BX,BUFFSTRING+2
608.    MOV AL,BUFFSTRING+1
609.    CBW
610.    MOV CX,AX
611.    ADD BX,AX
612.    MOV BYTE PTR [BX], '$'
613.    ;显示原字符串
614.    LEA DX,TIPOriginalString
615.    MOV AH,9
616.    INT 21H
617.    LEA DX,BUFFSTRING+2
618.    MOV AH,9
619.    INT 21H
620.    MOV DL,10
621.    MOV AH,02H
622.    INT 21H
623.    ;输出最大值提示符
624.    LEA DX,TIPMax
625.    MOV AH,9

```

```

626.    INT 21H
627.    ;找出最大值
628.    LEA BX,BUFFSTRING+2
629.    MOV DL,0
630.    FindMax:
631.        CMP [BX],DL
632.        JB NotBigger
633.        MOV DL,[BX]
634.        NotBigger:
635.            INC BX
636.    LOOP FindMax
637.    ;输出最大
638.    MOV AH,02H
639.    INT 21H
640.
641.    POP DI
642.    POP SI
643.    POP BP
644.    POP SP
645.    POP DX
646.    POP CX
647.    POP BX
648.    POP AX
649.    RET
650. FindMaxStr ENDP
651. ;把字符变为大写
652. ;入口参数: 无
653. ;出口参数: 无
654. CapitalizeString PROC FAR
655.    PUSH AX
656.    PUSH BX
657.    PUSH CX
658.    PUSH DX
659.    PUSH SP
660.    PUSH BP
661.    PUSH SI
662.    PUSH DI
663.
664.    ;清屏
665.    MOV AH,0
666.    MOV AL,3
667.    MOV BL,0
668.    INT 10H
669.
670.    MOV AH,2

```



```

671.    MOV BH,1
672.    MOV DL,0
673.    MOV DH,0
674.    INT 10H
675.    ;输入字符串
676.    CALL InputString
677.    ;清屏
678.    MOV AH,0
679.    MOV AL,3
680.    MOV BL,0
681.    INT 10H
682.    ;把输入字符串个数转化为字类型给 CX 来循环处理
683.    LEA BX,BUFFSTRING+2
684.    MOV AL,BUFFSTRING+1
685.    CBW
686.    MOV CX,AX
687.    ADD BX,AX
688.    MOV BYTE PTR [BX], '$'
689.    ;显示原字符串
690.    LEA DX,TIPOriginalString
691.    MOV AH,9
692.    INT 21H
693.    LEA DX,BUFFSTRING+2
694.    MOV AH,9
695.    INT 21H
696.
697.    MOV DL,10
698.    MOV AH,02H
699.    INT 21H
700.    ;转换为大写
701.    LEA BX,BUFFSTRING+2
702.    Capitalize:
703.        CMP BYTE PTR [BX],65;判断是字母还是数字
704.        JB NoCapitalize
705.        AND BYTE PTR [BX],5FH
706.        NoCapitalize:
707.            INC BX
708.    LOOP Capitalize
709.    ;;显示新字符串
710.    LEA DX,TIPNewString
711.    MOV AH,9
712.    INT 21H
713.    LEA DX,BUFFSTRING+2
714.    MOV AH,9
715.    INT 21H

```

```

716.
717.
718.     POP DI
719.     POP SI
720.     POP BP
721.     POP SP
722.     POP DX
723.     POP CX
724.     POP BX
725.     POP AX
726.     RET
727. CapitalizeString ENDP
728. ;输入字符串
729. InputString PROC FAR
730.     INPUTStr:
731.         LEA DX,TIPInputString
732.         MOV AH,9
733.         INT 21H
734.         LEA DX,BUFFSTRING
735.         MOV AH,0AH
736.         INT 21H
737.         CMP [BUFFSTRING+1],0
738.         JZ INPUTStr
739.         MOV DL,10
740.         MOV AH,02H
741.         INT 21H
742.         RET
743. InputString ENDP
744.
745. CODES ENDS
746.     END START

```