- 计科 180235 20188068 孔天欣 -

《汇编语言程序设计》大作业（2019-2020 年度）

1. 编写程序。从键盘输入一个两位十进制数，输出其二进制结果。

代码：

```
;---------- 题 1 ---------------

;编写程序。从键盘输入一个两位十进制数，
;输出其二进制结果。

;----------- 示 例 --------------
;输入：5
;输出：00000101

;输入：34
;输出：00100010

;输入:340
;输出:Only Two Digits Are Allowed

;输入:Q1
;输出：Not A Number

;----------------------------
DATAS SEGMENT
    ;字符串形式的十进制数字
    NUMSTR DB 3 DUP(?)
    ;防止 NUMSTR 过度填充
    BUFF_USELESS DB 32 DUP(0)
    WRONG_INFO DB 'Not A Number!$'
    WRONG_INFO_OVER DB 'Only Two Digits Are Allowed!$'
    ;字符转数字的变量
    NUMBER DB 0
    INPUT_INFO DB 'Please enter 2 decimal digits: $'
    OUTPUT_INFO DB 'The Binary number is: $'
    TIPS_INFO DB 'Do You Want To Repeat Again ?[y/n] $'
DATAS ENDS
```

```
33.
34. STACKS SEGMENT
35.     ;存放二进制位数的堆栈
36.     DW 16 DUP(?)
37. STACKS ENDS
38.
39. CODES SEGMENT
40.     ASSUME CS:CODES,DS:DATAS,SS:STACKS
41. START:
42. ;-----------------------------------MACRO
43.
44. ;宏，输出字符
45. PRINTC MACRO STR
46.      PROTECT_STACK
47.
48.      MOV AH,2
49.      MOV DL,STR
50.      INT 21H
51.
52.      RECOVER_STACK
53. ENDM
54.
55. ;宏，输出字符串
56. PRINTS MACRO STR
57.      PROTECT_STACK
58.      PUSH DS
59.
60.      MOV AX,SEG STR
61.      MOV DS,AX
62.      LEA DX,STR
63.      MOV AH,9
64.      INT 21H
65.
66.      POP DS
67.      RECOVER_STACK
68. ENDM
69.
70. ;宏，保护现场
71. PROTECT_STACK MACRO
72.     PUSH AX
73.     PUSH BX
74.     PUSH CX
75.     PUSH DX
76. ENDM
```

```
77.
78. ;宏，现场恢复
79. RECOVER_STACK MACRO
80.     POP DX
81.     POP CX
82.     POP BX
83.     POP AX
84.
85. ENDM
86.
87. ;宏，换行
88. NEXTLINE MACRO
89.     PRINTC 0AH
90.     PRINTC 0DH
91. ENDM
92.
93. ;宏 清屏
94. CLEAR_SCREEN MACRO
95.
96.     PROTECT_STACK
97.
98.     MOV AH,6
99.     MOV AL,0
100.     MOV BH,07H  ;黑底白字
101.     MOV CH,0    ;左上(0,0)
102.     MOV CL,0
103.     MOV DH,24
104.     MOV DL,79   ;右下(24,79)->25 * 80
105.     INT 10H
106.     ;移动光标到开头
107.     MOV AH,2
108.     MOV DH,0
109.     MOV DL,0
110.     MOV BH,0
111.     INT 10H
112.
113.     RECOVER_STACK
114.
115. ENDM
116.
117. ;宏，退出程序
118. RTSYS MACRO
119.     MOV AH,4CH
120.     INT 21H
```

```
121.
122. ENDM
123.
124. ;宏，用于反复执行程序
125. DO_AGAIN MACRO
126.     LOCAL RESTART
127.
128.     PRINTS TIPS_INFO
129.
130.     MOV AH,1H
131.     INT 21H
132.
133.     CMP AL,0DH
134.     JE RESTART
135.     CMP AL,'y'
136.     JE RESTART
137.     CMP AL,'Y'
138.     JE RESTART
139.     RTSYS
140.
141.    RESTART:
142.     CLEAR_SCREEN
143.     JMP STARTS
144.
145. ENDM
146.
147.
148.
149. ;-------------------------------------START
150.    STARTS:
151.     MOV AX,DATAS
152.     MOV DS,AX
153.     MOV AX,STACKS
154.     MOV SS,AX
155.
156.     ;数据重置
157.     CALL RESET_BEGIN
158.
159.     ;输出提示消息
160.     PRINTS INPUT_INFO
161.
162.    ;输入 2 位以内十进制数字
163.    INPUT:
164.     MOV AH,1
```

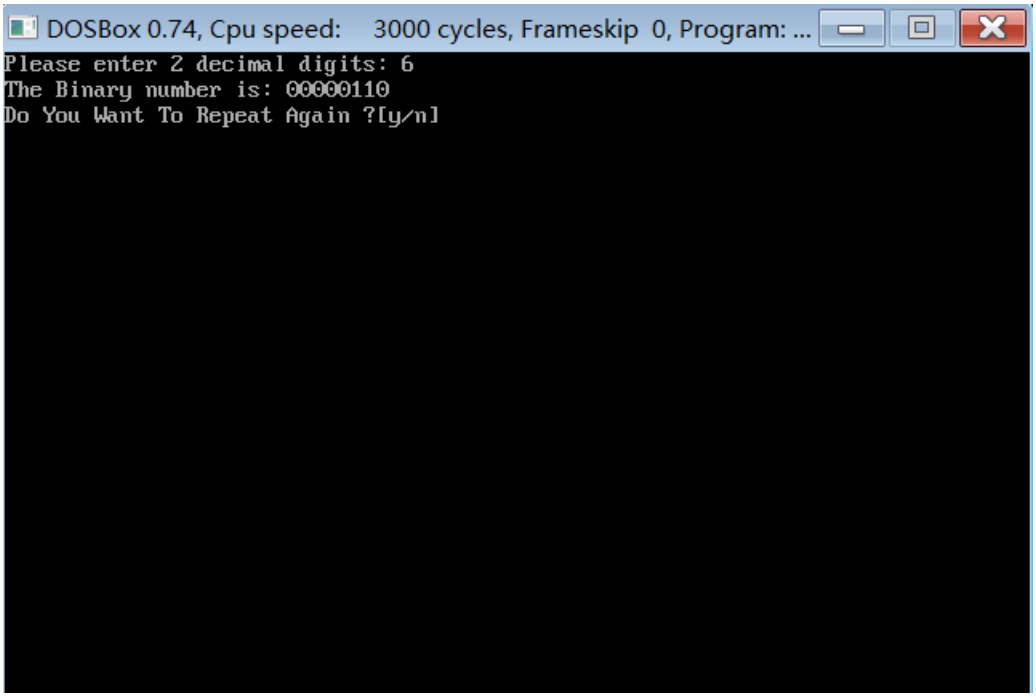
```
    INT 21H
    MOV NUMSTR[SI],AL
    INC SI
    CMP AL,0DH
    JNE INPUT

    ;判断字符是否是数字,如果不是报错
    CALL JUDGE_IS_NUM

    ;转为十进制数字
    TOD:
     ;SI:1->2 位
     MOV SI,1
     MOV BL,1
    TODIN:
     CMP SI,-1
     JE TOB
     ;是个位数?
     MOV AL,NUMSTR[SI]
     DEC SI
     CMP AL,0DH
     JE TODIN
     ;逐位相乘
     SUB AL,30H
     MUL BL
     ADD NUMBER,AL
     MOV AL,BL
     MOV BH,10
     MUL BH
     MOV BL,AL
     JMP TODIN

    ;转为二进制
    TOB:
     MOV AL,NUMBER
     MOV CX,0
    TOBIN:
     MOV BL,2
     CMP AL,0
     JE FILL
     MOV AH,0
     DIV BL
     MOV BL,AL
     MOV AL,0
```

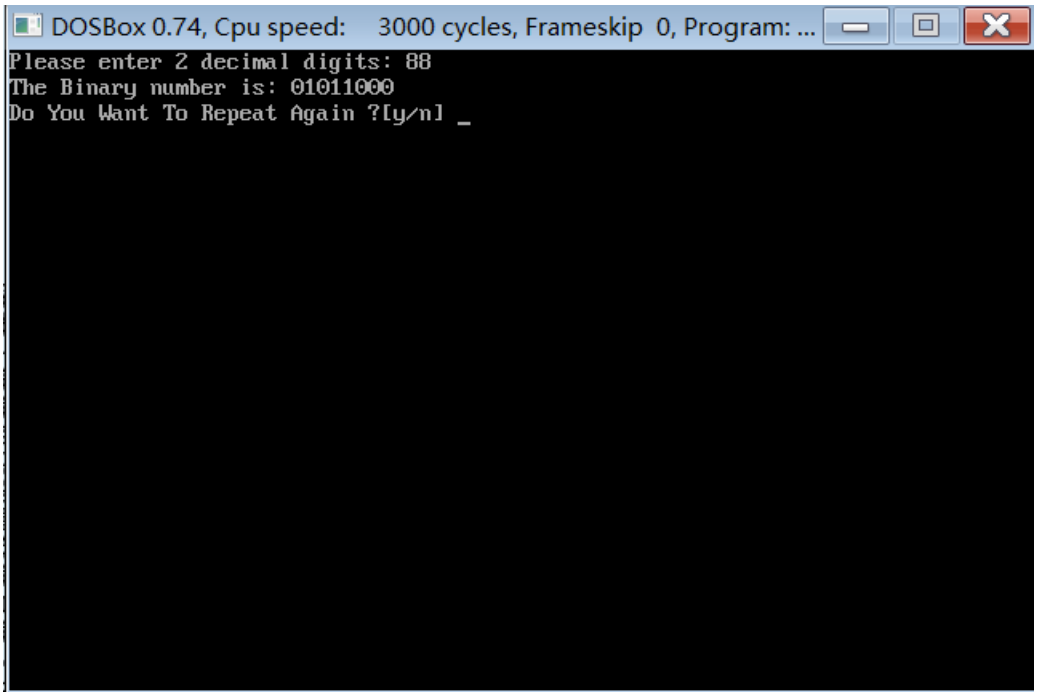
```
209.      ;余数入栈，商进入下一次循环
210.      PUSH AX
211.      INC CX
212.      MOV AL,BL
213.      JMP TOBIN
214.
215.     ;8 位补足
216.     FILL:
217.      MOV BX,8
218.      SUB BX,CX
219.      MOV CX,BX
220.     FILLIN:
221.      MOV AX,0
222.      PUSH AX
223.      LOOP FILLIN
224.
225.      ;输出提示信息
226.      PRINTS OUTPUT_INFO
227.
228.      MOV CX,8
229.     ;弹出堆栈，输出各个余数
230.     SHOW:
231.      POP AX
232.      MOV DL,AH
233.      ADD DL,30H
234.      PRINTC DL
235.     LOOP SHOW
236.      NEXTLINE
237.      JMP LETEND
238.
239.    ;非数字，输出错误信息
240.    ERROR:
241.     PRINTS WRONG_INFO
242.     NEXTLINE
243.     JMP LETEND
244.    ;字符串长度大于 2，报错
245.    ERROR_OVER:
246.     PRINTS WRONG_INFO_OVER
247.     NEXTLINE
248.
249.    ;重复程序
250.    LETEND:
251.     DO_AGAIN
252. ;---------------------------------------PROC
```

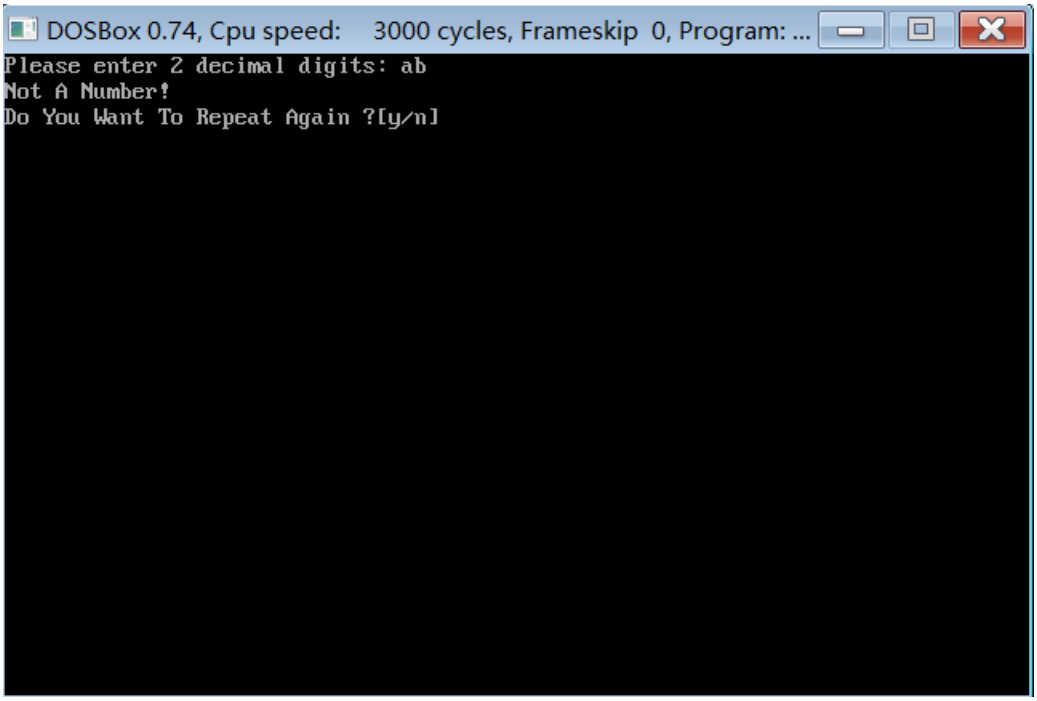
```
253.
254.  ;判断字符是否是数字,如果不是报错
255. JUDGE_IS_NUM PROC
256.
257.     PROTECT_STACK
258.
259.     ;只允许输入 2 位，否则直接报错
260.     CMP SI,3
261.     JA ERROR_OVER
262.
263.     MOV SI,0
264.     MOV CX,2
265.    JUDGE:
266.     MOV AL,NUMSTR[SI]
267.     CMP SI,0
268.     JNE JUDGE_IN
269.     ;第一个字符是回车？
270.     CMP AL,0DH
271.     JE ERROR
272.    JUDGE_IN:
273.     INC SI
274.     ;是回车？
275.     CMP AL,0DH
276.     JE TOD
277.     ;是 0~9 的数字？
278.     CMP AL,'0'
279.     JB ERROR
280.     CMP AL,'9'
281.     JA ERROR
282.     LOOP JUDGE
283.
284.     RECOVER_STACK
285.
286.     RET
287.
288. JUDGE_IS_NUM ENDP
289.
290. ;数据重置
291. RESET_BEGIN PROC
292.     MOV AX,0
293.     MOV SI,0
294.     MOV NUMBER,0
295.     RET
296. RESET_BEGIN ENDP
```

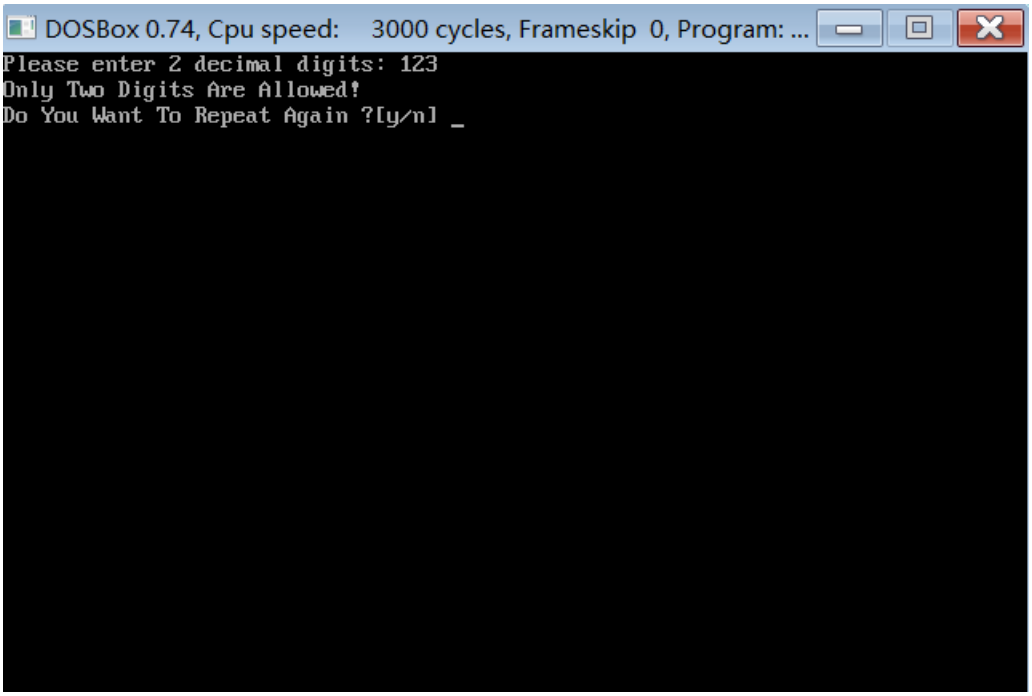
```
297.
298. CODES ENDS
299.     END START
```

运行截图：









2. 编写程序。从键盘输入一串字符，按照其 ASCII 码从大到小的顺序排列，并把排序后的结果显示出来。

代码：

```
1.  ;---------- 题 2 ---------------
2.
```

```
;编写程序。从键盘输入一串字符，
;按照其 ASCII 码从大到小的顺序排列，
;并把排序后的结果显示出来。

;---------- 示 例 --------------

;输入：--==//..123044qwertyuiop
;输出：ywutrqpoie==443210//..--

;----------------------------
DATAS SEGMENT
    INPUTINFO DB 'Please Input The String:$'
    OUTPUTINFO DB 'After Sorted The Answer Is:$'
    ;输入的字符串
    STRING DB 64,?,64 DUP('$')
    ;NUM=STRING[1]，获取实际输入个数
    NUM DW 0
    TIPS_INFO DB 'Do You Want To Repeat Again ?[y/n] $'
DATAS ENDS

STACKS SEGMENT
    DW 32 DUP(?)
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:

;---------------------------------------MACRO

;宏，输出字符
PRINTC MACRO STR
     PROTECT_STACK

     MOV AH,2
     MOV DL,STR
     INT 21H

     RECOVER_STACK
ENDM

;宏，输出字符串
PRINTS MACRO STR
     PROTECT_STACK
```

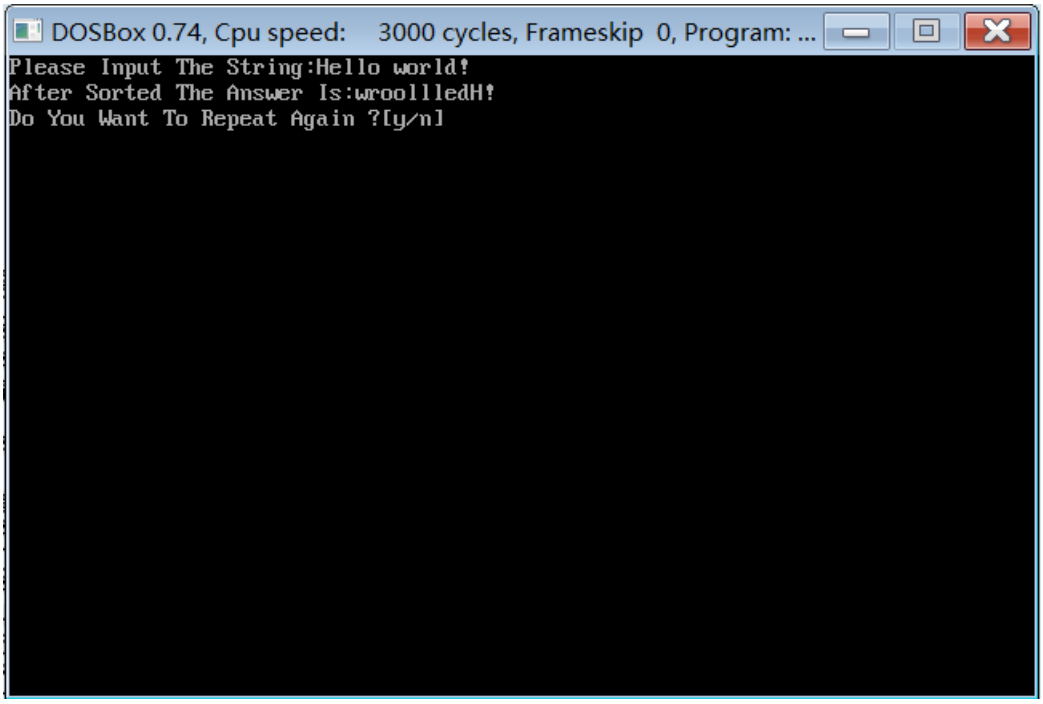
```
47.      PUSH DS
48.
49.      MOV AX,SEG STR
50.      MOV DS,AX
51.      LEA DX,STR
52.      MOV AH,9
53.      INT 21H
54.
55.      POP DS
56.      RECOVER_STACK
57. ENDM
58.
59. ;宏，保护现场
60. PROTECT_STACK MACRO
61.     PUSH AX
62.     PUSH BX
63.     PUSH CX
64.     PUSH DX
65. ENDM
66.
67. ;宏，现场恢复
68. RECOVER_STACK MACRO
69.     POP DX
70.     POP CX
71.     POP BX
72.     POP AX
73.
74. ENDM
75.
76. ;宏，换行
77. NEXTLINE MACRO
78.     PRINTC 0AH
79.     PRINTC 0DH
80. ENDM
81.
82. ;宏 清屏
83. CLEAR_SCREEN MACRO
84.
85.     PROTECT_STACK
86.
87.     MOV AH,6
88.     MOV AL,0
89.     MOV BH,07H  ;黑底白字
90.     MOV CH,0    ;左上(0,0)
```

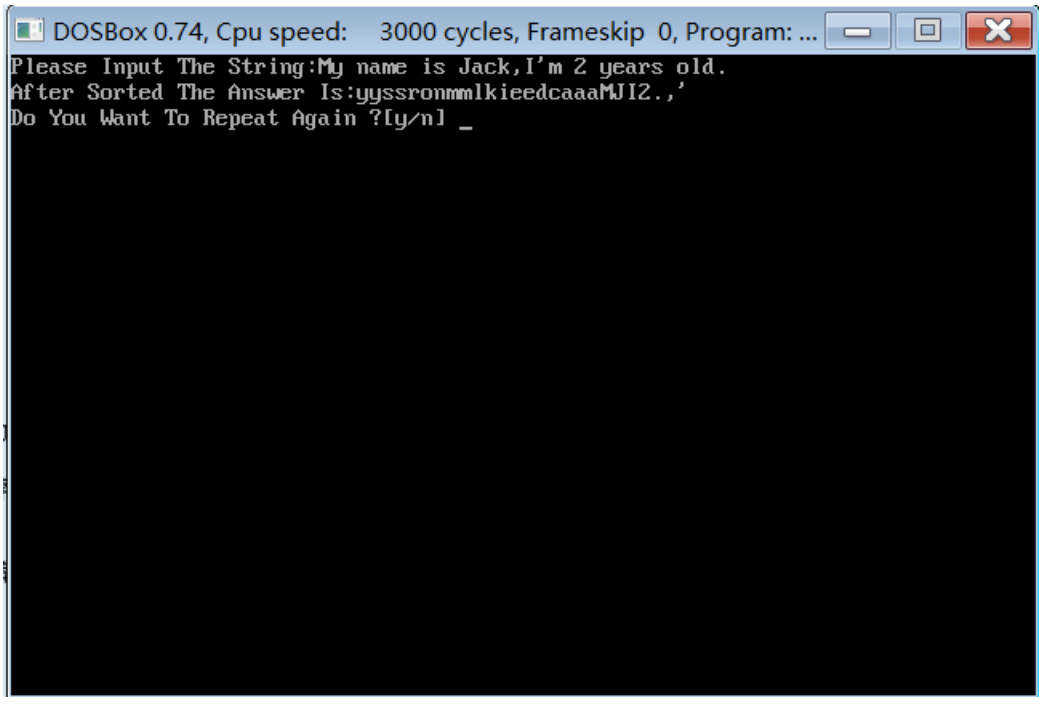
```
    MOV CL,0
    MOV DH,24
    MOV DL,79   ;右下(24,79)->25 * 80
    INT 10H
    ;移动光标到开头
    MOV AH,2
    MOV DH,0
    MOV DL,0
    MOV BH,0
     INT 10H

     RECOVER_STACK

ENDM

;宏，退出程序
RTSYS MACRO
    MOV AH,4CH
    INT 21H

ENDM

;宏，用于反复执行程序
DO_AGAIN MACRO
    LOCAL RESTART

    PRINTS TIPS_INFO

    MOV AH,1H
    INT 21H

    CMP AL,0DH
    JE RESTART
    CMP AL,'y'
    JE RESTART
    CMP AL,'Y'
    JE RESTART
    RTSYS

   RESTART:
    CLEAR_SCREEN
    JMP STARTS

ENDM
```

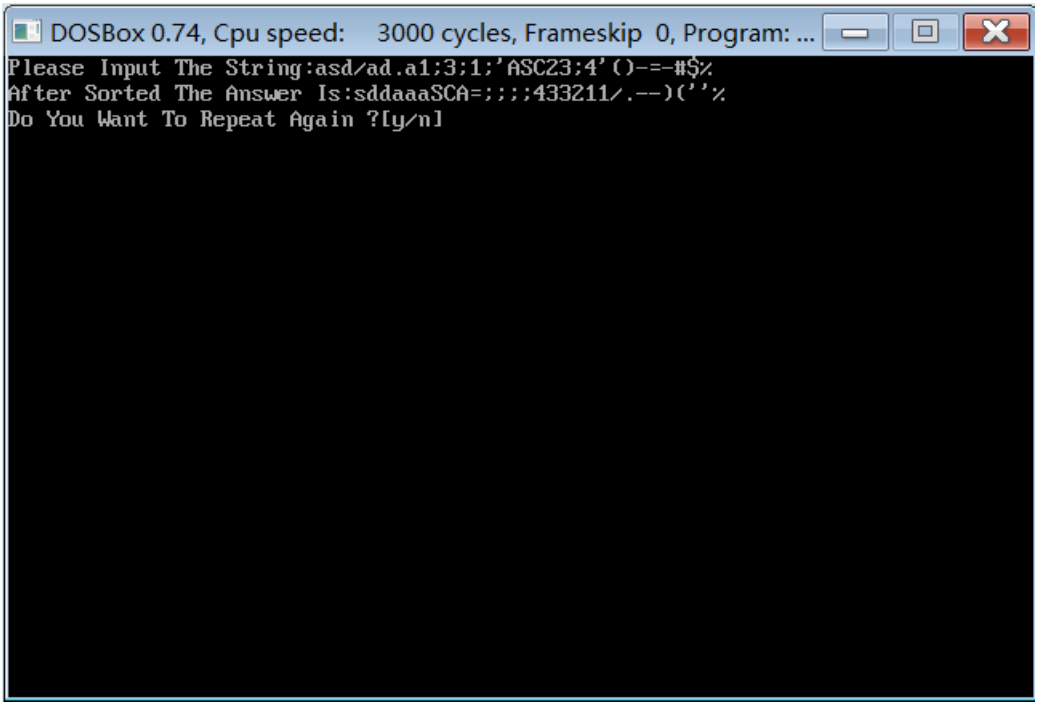
```
135.
136. ;-----------------------------------------START
137.    STARTS:
138.     MOV AX,DATAS
139.     MOV DS,AX
140.     MOV AX,STACKS
141.     MOV SS,AX
142.
143.     ;数据重置
144.     CALL RESET_BEGIN
145.
146.     ;输入提示
147.     PRINTS INPUTINFO
148.
149.     ;输入字符串
150.     MOV SI,1
151.     LEA DX,STRING
152.     MOV AH,10
153.     INT 21H
154.     ;取字节个数
155.     MOV AL,STRING[SI]
156.     MOV AH,0
157.     MOV NUM,AX
158.
159.     ;调用冒泡排序
160.     CALL SORT
161.
162.     ;回车换行
163.     NEXTLINE
164.     ;输出提示
165.     PRINTS OUTPUTINFO
166.     ;输出 ASCII 排序后的字符串
167.     PRINTS STRING[2]
168.     NEXTLINE
169.
170.     DO_AGAIN
171.
172. ;-------------------------------------------------PROC
173.
174. ;开始冒泡排序
175. SORT PROC
176.
177.      CMP NUM,1
178.      JE DO_END
```

```
179.
180.      ;现场保护
181.      PROTECT_STACK
182.
183.      MOV CX,NUM
184.      ;CX-1 趟
185.      DEC CX
186.     LOOP1:
187.      PUSH CX
188.      MOV SI,0
189.     LOOP2:
190.      MOV AL,STRING[2+SI]
191.      CMP AL,STRING[2+SI+1]
192.      ;逆序排序
193.      JGE NEXT
194.      XCHG AL,STRING[2+SI+1]
195.      MOV STRING[2+SI],AL
196.     NEXT:
197.      ADD SI,1
198.      LOOP LOOP2
199.      POP CX
200.      LOOP LOOP1
201.      ;恢复现场
202.      RECOVER_STACK
203.     DO_END:
204.      RET
205. SORT ENDP
206.
207. ;数据重置
208. RESET_BEGIN PROC
209.
210.     MOV CX,64
211.     MOV SI,0
212.    RESET:
213.     MOV STRING[SI],'$'
214.     INC SI
215.     LOOP RESET
216.     RET
217.
218. RESET_BEGIN ENDP
219.
220. CODES ENDS
221.     END START
```

运行截图：


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
Please Input The String:Hello world!
After Sorted The Answer Is:wroollledH!
Do You Want To Repeat Again ?[y/n]


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
Please Input The String:My name is Jack,I'm 2 years old.
After Sorted The Answer Is:yyssronmmlkieedcaaaMJI2.,'
Do You Want To Repeat Again ?[y/n] _



3. 编写 90H 号软件中断程序，在第（10）行（班级序列号 35）列彩色显示你的姓名(注：要求中断驻留，行从第 0 行开始计算，列从第 0 列开始计算)。

代码：

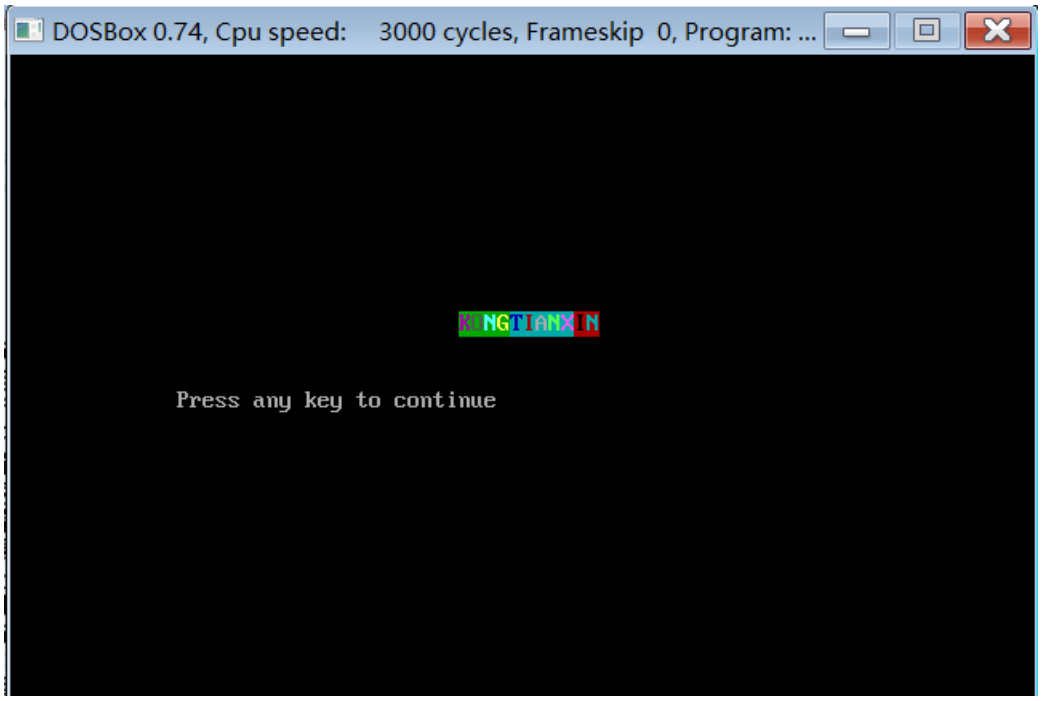
```
;---------- 题 3 ---------------

;3. 编写 90H 号软件中断程序，
;在第（10）行**（班级序列号为 35）**列彩色显示你的姓名
;(注：要求中断驻留，行从第 0 行开始计算，列从第 0 列开始计算)。

;-----------------------------

DATAS SEGMENT
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:

;-------------------------------------MACRO
```

```
20.
21. ;宏，保护现场
22. PROTECT_STACK MACRO
23.     PUSH AX
24.     PUSH BX
25.     PUSH CX
26.     PUSH DX
27. ENDM
28.
29. ;宏，现场恢复
30. RECOVER_STACK MACRO
31.     POP DX
32.     POP CX
33.     POP BX
34.     POP AX
35.
36. ENDM
37.
38. ;宏 清屏
39. CLEAR_SCREEN MACRO
40.
41.     PROTECT_STACK
42.
43.     MOV AH,6
44.     MOV AL,0
45.     MOV BH,07H  ;黑底白字
46.     MOV CH,0    ;左上(0,0)
47.     MOV CL,0
48.     MOV DH,24
49.     MOV DL,79   ;右下(24,79)->25 * 80
50.     INT 10H
51.     ;移动光标到开头
52.     MOV AH,2
53.     MOV DH,0
54.     MOV DL,0
55.     MOV BH,0
56.     INT 10H
57.
58.     RECOVER_STACK
59.
60. ENDM
61. ;---------------------------------------START
62.
63.     ;将 90H 号中断放入向量表中
```

```
    LEA DX,OUTPUTSTART
    MOV AX,SEG OUTPUTSTART
    MOV DS,AX
    MOV AL,90H
    MOV AH,25H
    INT 21H

    ;中断驻留
    MOV AH,31H
    MOV AL,0
    MOV DX,OUTPUTEND-OUTPUTSTART+16

    ;触发 90H 号中断
    INT 90H

    MOV AH,4CH
    INT 21H

    ;中断程序
    OUTPUTSTART:

    JMP CODE
    ;姓名
    MYNAME DB 'KONGTIANXIN'
    ;颜色
    COLOR DB 1EH
    CODE:

    ;现场保护
    PUSH DS
    PROTECT_STACK
    PUSH CS
    POP DS

    CLEAR_SCREEN

     ;移动光标到 10 行 35 列
     MOV AH,2
     MOV DH,10
     MOV DL,35
     MOV BH,0
     INT 10H

     ;写显存
```

```
    MOV BX,0B800H
    MOV ES,BX
    ;行：A0H * AH = 640H
    MOV BX,640H
    ;列 2H * 23H(35D) = 46H
    MOV DI,46H
    ;名字的长度作为循环
    MOV CX,DWORD PTR COLOR-MYNAME
    MOV SI,0
    PAINT:
    ;输出字符
    MOV AH,2
    MOV DL,MYNAME[SI]
    INC SI
    INT 21H
    MOV AL,COLOR
    ;填色，奇地址存放属性
    ADD BYTE PTR ES:[BX+1+DI],AL
    ;下一个字符换颜色
    ADD AL,3H
    MOV COLOR,AL
    ;下一个相邻位置
    ADD BX,2
    LOOP PAINT

    ;现场恢复
    RECOVER_STACK
    POP DS
    IRET
    OUTPUTEND:NOP
CODES ENDS
    END START
```

运行截图：



4. 编写程序。从键盘输入多个带符号数：

   ①统计负数的个数；

   ②计算所有负数之和，并显示出十进制结果。

   代码：

```
;---------- 题 4 ---------------

;编写程序。从键盘输入多个带符号数：
;①统计负数的个数；
;②计算所有负数之和，并显示出十进制结果。

;---------- 示 例 --------------


;输入：34 -2 -3 -4 43
;输出：The Number Of Negative Numbers Is : 3
;      The Sum Of Negative Numbers Is : -9


;输入:0 -200 -100 -50000 50000
;输出：The Number Of Negative Numbers Is : 3
;      The Sum Of Negative Numbers Is : -50300

;输入:0 200 100 50000 50000
```

```
;输出：The Number Of Negative Numbers Is : 0
;       The Sum Of Negative Numbers Is : 0

;输入：23 33 5G
;输出：Please Input Correct Synatax

;----------------------------

DATAS SEGMENT
    ;存放数字字符串，以$分割
    ARRAYSTR DB 128 DUP('$')
    ;ARRAYSTR 的实际最大下标
    PTRNUM DW 0
    NUM_OF_NEGA DB 0
    ;临时数字，用于暂存十进制数（无论正负）
    TEMP DW 0
    ;是否是负数的标志
    SIGN DB 0
    ;负数的总和,上限 65535
    SUM DW 0
    INPUT_INFO DB 'Please Input The Number List: $'
    ;错误信息：请输入正确的格式
    WRONG_INFO DB 'Please Input Correct Synatax!$'
    NUM_INFO DB 'The Number Of Negative Numbers Is : $'
    SUM_INFO DB 'The Sum Of Negative Numbers Is : $'
    OVER_INFO DB 'Overflow! $'
    TIPS_INFO DB 'Do You Want To Repeat Again ?[y/n] $'
DATAS ENDS

STACKS SEGMENT
STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:

;--------------------------------MACRO

;宏，输出字符
PRINTC MACRO STR
     PROTECT_STACK

     MOV AH,2
     MOV DL,STR
```

```
64.      INT 21H
65.
66.      RECOVER_STACK
67. ENDM
68.
69. ;宏，输出字符串
70. PRINTS MACRO STR
71.      PROTECT_STACK
72.      PUSH DS
73.
74.      MOV AX,SEG STR
75.      MOV DS,AX
76.      LEA DX,STR
77.      MOV AH,9
78.      INT 21H
79.
80.      POP DS
81.      RECOVER_STACK
82. ENDM
83.
84. ;宏，保护现场
85. PROTECT_STACK MACRO
86.     PUSH AX
87.     PUSH BX
88.     PUSH CX
89.     PUSH DX
90. ENDM
91.
92. ;宏，现场恢复
93. RECOVER_STACK MACRO
94.     POP DX
95.     POP CX
96.     POP BX
97.     POP AX
98.
99. ENDM
100.
101. ;宏，换行
102. NEXTLINE MACRO
103.     PRINTC 0AH
104.     PRINTC 0DH
105. ENDM
106.
107. ;宏 清屏
```

```
CLEAR_SCREEN MACRO

    PROTECT_STACK

    MOV AH,6
    MOV AL,0
    MOV BH,07H  ;黑底白字
    MOV CH,0    ;左上(0,0)
    MOV CL,0
    MOV DH,24
    MOV DL,79   ;右下(24,79)->25 * 80
    INT 10H
    ;移动光标到开头
    MOV AH,2
    MOV DH,0
    MOV DL,0
    MOV BH,0
    INT 10H

    RECOVER_STACK

ENDM

;宏，退出程序
RTSYS MACRO
    MOV AH,4CH
    INT 21H

ENDM

;宏，用于反复执行程序
DO_AGAIN MACRO
    LOCAL RESTART

    PRINTS TIPS_INFO

    MOV AH,1H
    INT 21H

    CMP AL,0DH
    JE RESTART
    CMP AL,'y'
    JE RESTART
    CMP AL,'Y'
```

```
    JE RESTART
    RTSYS

   RESTART:
    CLEAR_SCREEN
    JMP STARTS

ENDM


;宏，输出数字的十进制字符串
PRINT_NUM MACRO NUM

   LOCAL DO_ING
   LOCAL DO_END

   ;保护现场
   PROTECT_STACK

   MOV AX,WORD PTR NUM
   MOV DX,0
   MOV BX,10
   DO_ING:
    DIV BX
    ;余数入栈，商进入下一次循环
    PUSH DX
    MOV DX,0
    INC CX
    CMP AX,0
    JE DO_END
    JMP DO_ING
   ;按位输出
   DO_END:
    POP AX
    ADD AX,3030H
    PRINTC AL
    LOOP DO_END

    ;现场恢复
    RECOVER_STACK

ENDM

;-----------------------------START
```

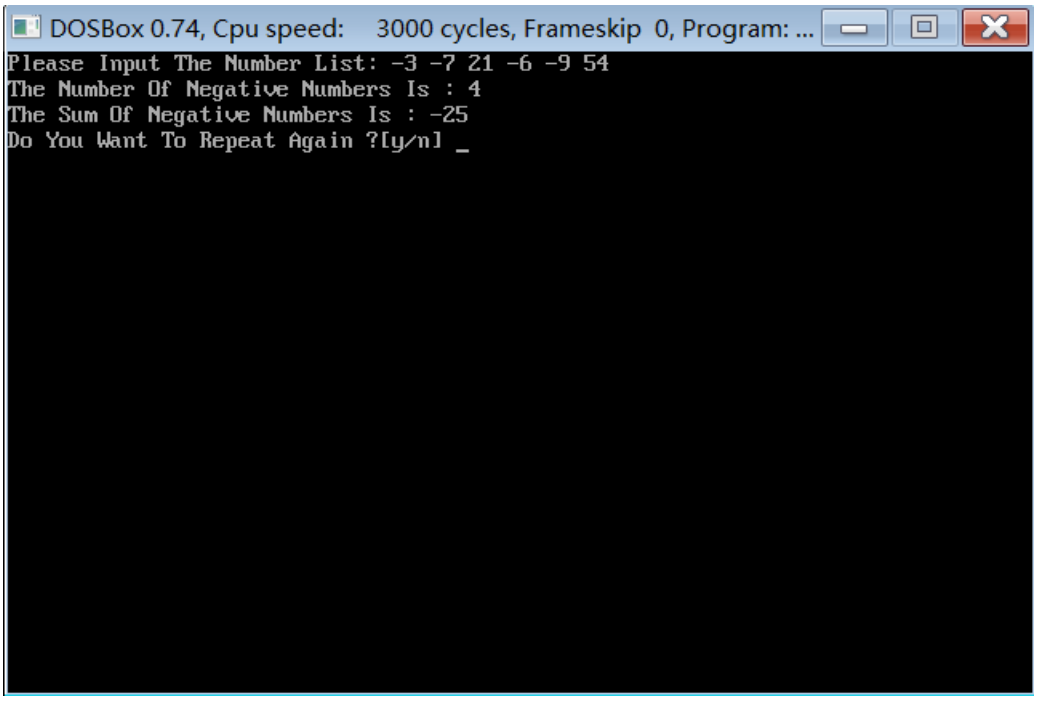
```
196.    STARTS:
197.     MOV AX,DATAS
198.     MOV DS,AX
199.
200.     ;重置数据
201.     CALL RESET_BEGIN
202.
203.     PRINTS INPUT_INFO
204.
205.     ;先输入一串带符号数，空格作为分割，
206.     ;在存储单元中以'$'代替
207.     INPUT:
208.      MOV AH,1
209.      INT 21H
210.
211.      ;是回车？
212.      CMP AL,0DH
213.      ;跳出循环
214.      JE TONUM
215.
216.      ;是空格？
217.      CMP AL,20H
218.      JNE ISNEGA
219.      ;分隔符
220.      INC SI
221.      JMP INPUT
222.
223.     ;是负号？
224.     ISNEGA:
225.      CMP AL,'-'
226.      JNE NOTNEGA
227.      MOV ARRAYSTR[SI],'-'
228.      INC SI
229.      JMP INPUT
230.     ;不是负号的情况下
231.     NOTNEGA:
232.      ;是 0~9 的数字？
233.      CMP AL,'0'
234.      ;不是，报错
235.      JB ERROR
236.      CMP AL,'9'
237.      JA ERROR
238.      MOV ARRAYSTR[SI],AL
239.      INC SI
```

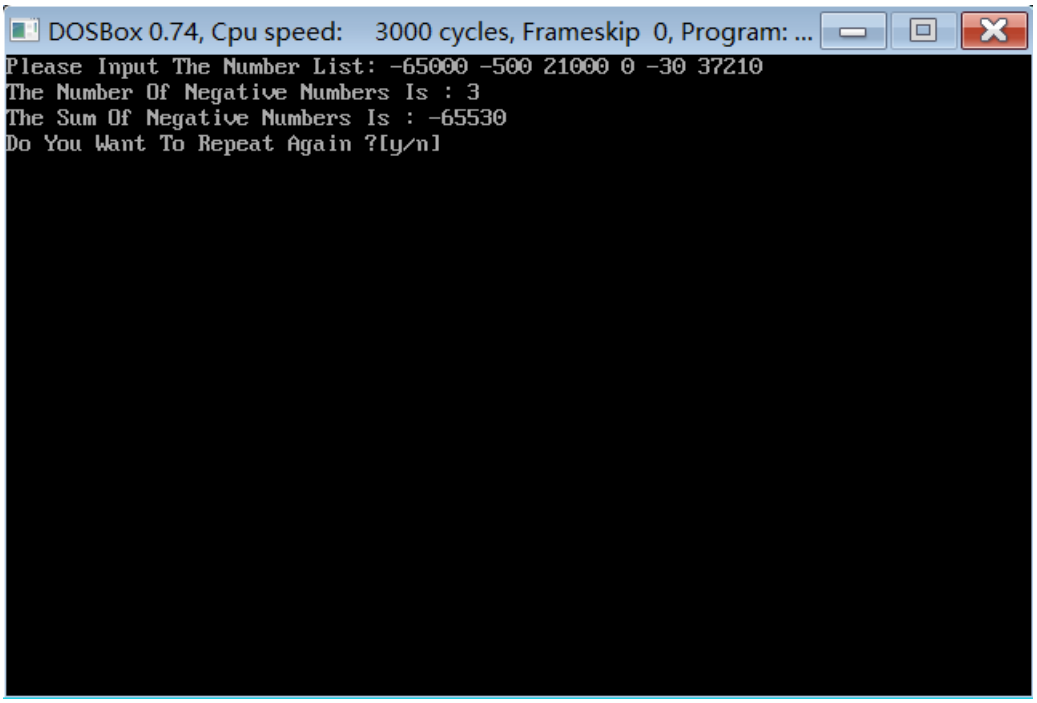
```
240.      JMP INPUT
241.
242.     ;所有负数相加，并统计负数个数
243.     TONUM:
244.      MOV PTRNUM,SI
245.      ;数组下标
246.      MOV SI,0
247.      ;一个数字除符号外的位数
248.      MOV CX,0
249.      ;乘数
250.      MOV BX,1
251.     TONUMIN:
252.
253.      ;如果数组下标已经大于 PTRNUM，跳出
254.      CMP PTRNUM,SI
255.      JB PRINT_ALL
256.
257.      MOV AL,ARRAYSTR[SI]
258.      CMP AL,'$'
259.      JE TOSUM
260.      CMP AL,'-'
261.      JE ADDNEGA
262.      ;是数字字符，入栈
263.      MOV AH,0
264.      PUSH AX
265.      INC SI
266.      INC CX
267.      JMP TONUMIN
268.     ADDNEGA:
269.      ;是负数
270.      MOV SIGN,1
271.      INC NUM_OF_NEGA
272.      INC SI
273.      JMP TONUMIN
274.     TOSUM:
275.      ;从个位数出栈
276.      POP AX
277.      SUB AL,30H
278.      MUL BX
279.      ADD TEMP,AX
280.      MOV AX,BX
281.      MOV BX,10
282.      MUL BX
283.      MOV BX,AX
```

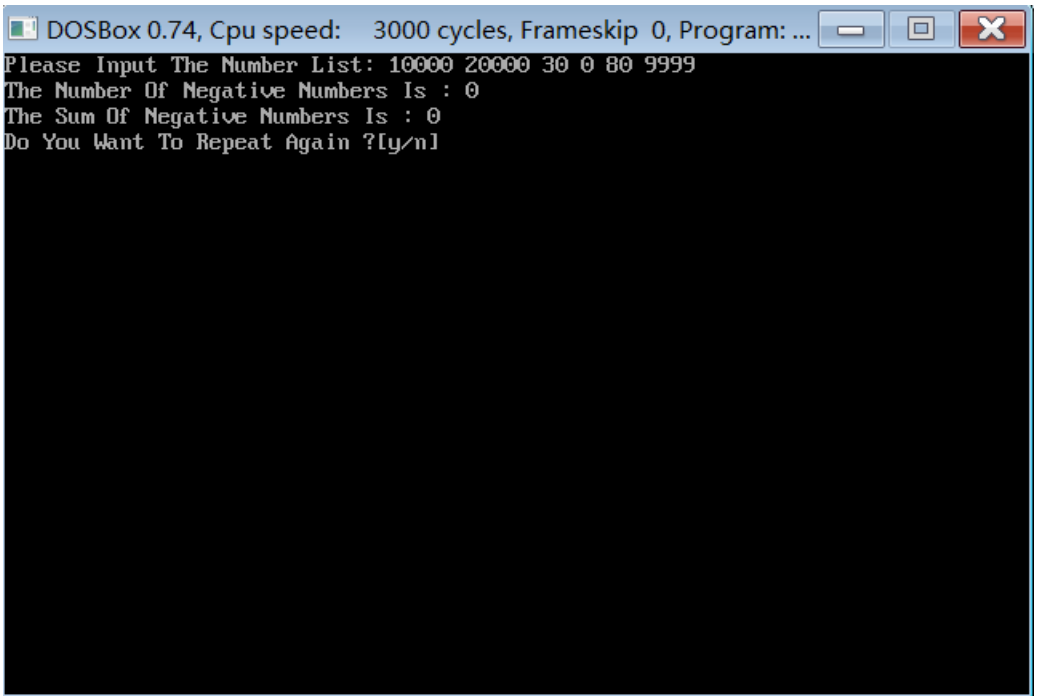
```
284.      LOOP TOSUM
285.      ;是负数吗?
286.      CMP SIGN,1
287.      JNE TOSUMIN
288.      ;是负数，SUM+=TEMP，重置
289.      MOV AX,TEMP
290.      ADD SUM,AX
291.      INC SI
292.      CALL RESET
293.      JMP TONUMIN
294.     ;不是负数，重置
295.     TOSUMIN:
296.      INC SI
297.      CALL RESET
298.      JMP TONUMIN
299.
300.     ;输出所有消息
301.     PRINT_ALL:
302.
303.      PRINTS NUM_INFO
304.      PRINT_NUM NUM_OF_NEGA
305.      NEXTLINE
306.
307.      PRINTS SUM_INFO
308.
309.      ;没有负数也就没必要打印符号
310.      CMP NUM_OF_NEGA,0
311.      JE DO_NOT_PRINTN
312.      PRINTC '-'
313.     DO_NOT_PRINTN:
314.      PRINT_NUM SUM
315.      NEXTLINE
316.
317.      JMP LETE
318.     ;输出错误消息
319.     ERROR:
320.      NEXTLINE
321.      PRINTS WRONG_INFO
322.      NEXTLINE
323.      JMP LETE
324.
325.     LETE:
326.
327.     DO_AGAIN
```

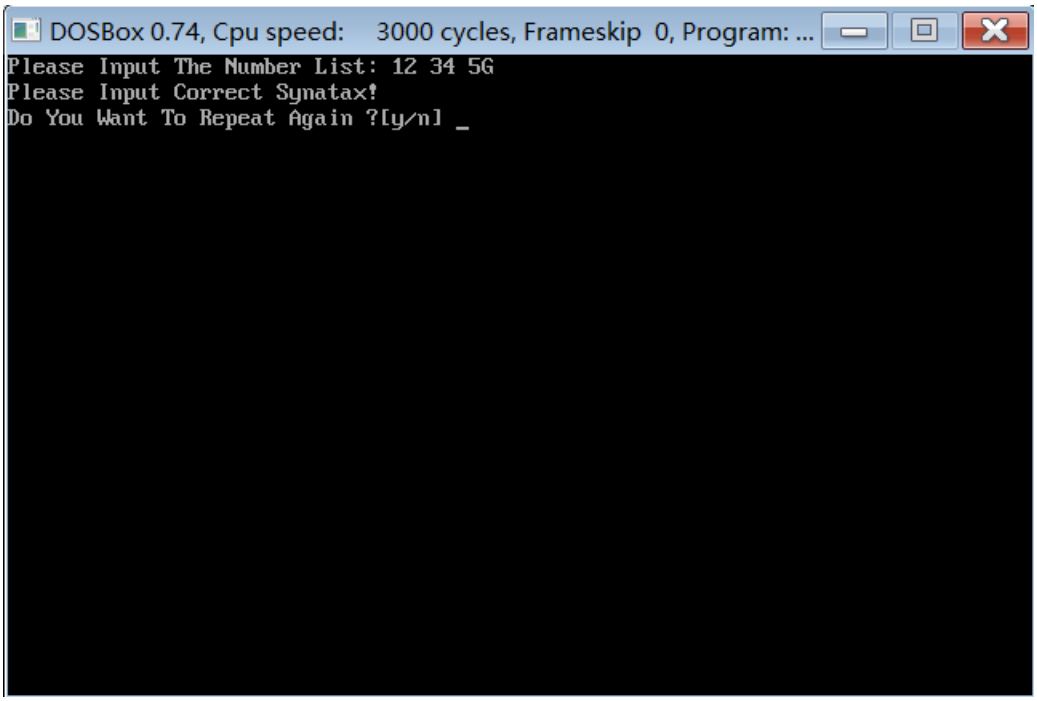
```
328.
329. ;--------------------------------PROC
330.
331. ;重置相关信息
332. RESET PROC
333.       MOV TEMP,0
334.       MOV BX,1
335.       MOV SIGN,0
336.       RET
337. RESET ENDP
338.
339. ;数据重置
340. RESET_BEGIN PROC
341.
342.      MOV SI,0
343.      MOV TEMP,0
344.      MOV SUM,0
345.      MOV NUM_OF_NEGA,0
346.      MOV CX,128
347.     RESET_CLEAR:
348.      MOV ARRAYSTR[SI],'$'
349.      INC SI
350.      LOOP RESET_CLEAR
351.      MOV SI,0
352.      RET
353.
354. RESET_BEGIN ENDP
355.
356. CODES ENDS
357.      END START
```

运行截图：


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
Please Input The Number List: -3 -7 21 -6 -9 54
The Number Of Negative Numbers Is : 4
The Sum Of Negative Numbers Is : -25
Do You Want To Repeat Again ?[y/n] _


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
Please Input The Number List: -65000 -500 21000 0 -30 37210
The Number Of Negative Numbers Is : 3
The Sum Of Negative Numbers Is : -65530
Do You Want To Repeat Again ?[y/n]





5. 读取系统时间的百分秒作为 0—99 之间的随机数，实现两个随机数的加、减、乘、除运算，并显示算式及计算结果。

代码：

```
;---------- 题 5 ---------------

;读取系统时间的百分秒作为 0—99 之间的随机数，
```

```
;实现两个随机数的加、减、乘、除运算，
;并显示算式及计算结果。

;-----------------------------

DATAS SEGMENT
    ;MTIME：存放百分秒数值
    MTIME_1 DW 0
    MTIME_2 DW 0
    NUM_INFO_BEFORE DB 'The Frist Number Is : $'
    NUM_INFO_AFTER DB 'The Second Number Is : $'
    ADD_INFO DB 'NUM1 + NUM2 = $'
    SUB_INFO DB 'NUM1 - NUM2 = $'
    MUL_INFO DB 'NUM1 * NUM2 = $'
    DIV_INFO DB 'NUM1 / NUM2 = $'
    MOD_INFO DB ' ,MOD = $'
    TIPS_INFO DB 'Do You Want To Repeat Again ?[y/n] $'
    DELAY_TIME DB 14
DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:

;-------------------------------------MACRO

;宏，输出字符
PRINTC MACRO STR
     PROTECT_STACK

     MOV AH,2
     MOV DL,STR
     INT 21H

     RECOVER_STACK
ENDM

;宏，输出字符串
PRINTS MACRO STR
     PROTECT_STACK
```

```
48.      PUSH DS
49.
50.      MOV AX,SEG STR
51.      MOV DS,AX
52.      LEA DX,STR
53.      MOV AH,9
54.      INT 21H
55.
56.      POP DS
57.      RECOVER_STACK
58. ENDM
59.
60. PROTECT_STACK MACRO
61.     PUSH AX
62.     PUSH BX
63.     PUSH CX
64.     PUSH DX
65. ENDM
66.
67. RECOVER_STACK MACRO
68.     POP DX
69.     POP CX
70.     POP BX
71.     POP AX
72.
73. ENDM
74.
75. ;宏，换行
76. NEXTLINE MACRO
77.     PRINTC 0AH
78.     PRINTC 0DH
79. ENDM
80.
81. ;宏 清屏
82. CLEAR_SCREEN MACRO
83.
84.     PROTECT_STACK
85.
86.     MOV AH,6
87.     MOV AL,0
88.     MOV BH,07H  ;黑底白字
89.     MOV CH,0    ;左上(0,0)
90.     MOV CL,0
91.     MOV DH,24
```

```
    MOV DL,79   ;右下(24,79)->25 * 80
    INT 10H
    ;移动光标到开头
    MOV AH,2
    MOV DH,0
    MOV DL,0
    MOV BH,0
    INT 10H

     RECOVER_STACK

ENDM

;宏，输出数字的十进制字符串
PRINT_NUMS MACRO NUM

    LOCAL DO_ING
    LOCAL DO_END

    ;保护现场
    PROTECT_STACK

    MOV AX,NUM
    MOV DX,0
    MOV CX,0
    MOV BX,10
    DO_ING:
     DIV BX
     ;余数入栈，商进入下一次循环
     PUSH DX
     MOV DX,0
     INC CX
     CMP AX,0
     JE DO_END
     JMP DO_ING
    ;按位输出
    DO_END:
     POP AX
     ADD AX,3030H
     PRINTC AL
     LOOP DO_END

     ;现场恢复
     RECOVER_STACK
```

```
136. ENDM
137.
138. ;宏，获取当前时间
139. GET_MTIME MACRO ARG,INFO
140.
141.     MOV AH,2CH      ;DL->millsecond
142.     INT 21H
143.     MOV DH,0
144.     MOV ARG,DX
145.
146.     PRINTS INFO
147.     PRINT_NUMS ARG
148.     NEXTLINE
149.
150. ENDM
151.
152. ;宏，退出程序
153. RTSYS MACRO
154.     MOV AH,4CH
155.     INT 21H
156.
157. ENDM
158.
159. ;宏，反复执行程序
160. DO_AGAIN MACRO
161.     LOCAL RESTART
162.
163.     PRINTS TIPS_INFO
164.
165.     MOV AH,1H
166.     INT 21H
167.
168.     CMP AL,0DH
169.     JE RESTART
170.     CMP AL,'y'
171.     JE RESTART
172.     CMP AL,'Y'
173.     JE RESTART
174.     RTSYS
175.
176.    RESTART:
177.     CLEAR_SCREEN
178.     JMP STARTS
179. ENDM
```

```
;-----------------------------------------中断

;---除 0(0 号)的中断改写-----

;将 00H 号中断放入向量表中
    LEA DX,DIVIDE_START
    MOV AX,SEG DIVIDE_START
    MOV DS,AX
    MOV AL,00H
    MOV AH,25H
    INT 21H

    ;中断驻留
    MOV AH,31H
    MOV AL,0
    MOV DX,DIVIDE_END-DIVIDE_START+16

    JMP STARTS

    ;中断程序
   DIVIDE_START:
    JMP CODE
    DIVIDE_ERROR DB '0 Cannot Be Used As A Divisor.So Please *RESTART* The Program.$'
   CODE:

    ;现场保护
    PUSH DS
    PUSH DX
    PUSH AX
    PUSH CS
    POP DS

    PRINTS DIVIDE_ERROR

    ;现场恢复
    POP AX
    POP DX
    POP DS

    ;回到界面
    MOV AX,4C00H
    INT 21H

```

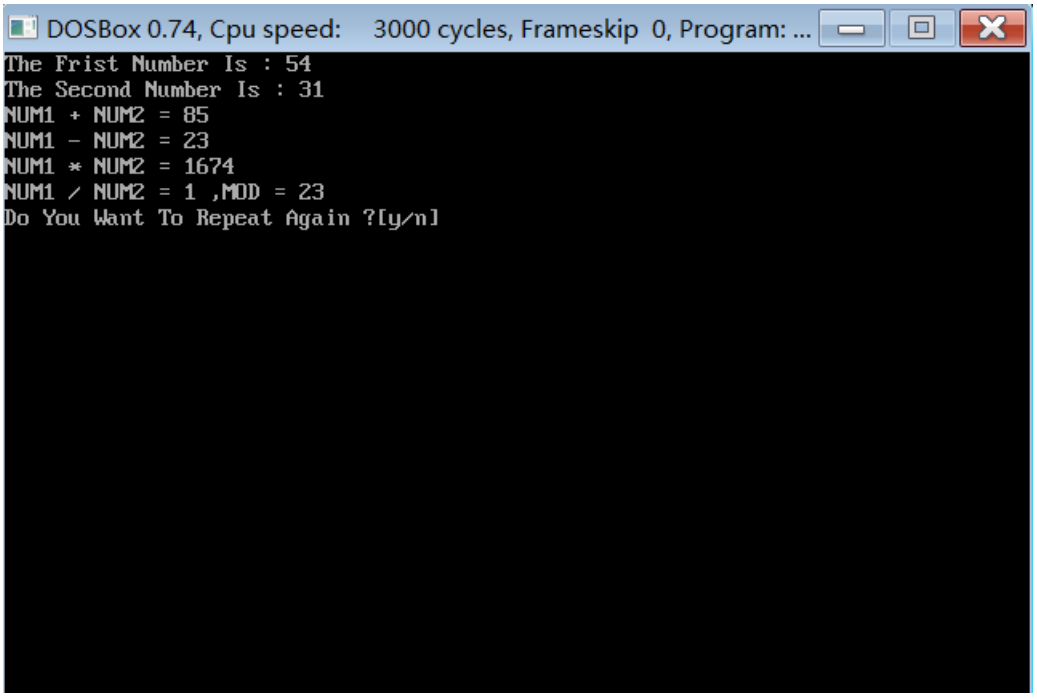
```
223.     DIVIDE_END:NOP
224.
225. ;-----------------------------------------START
226.    STARTS:
227.     MOV AX,DATAS
228.     MOV DS,AX
229.
230.     ;MTIME_1 = 当前百分秒
231.     GET_MTIME MTIME_1,NUM_INFO_BEFORE
232.
233.     ;执行延时
234.     CALL DELAY
235.
236.     ;MTIME_1 = 延时后的百分秒
237.     GET_MTIME MTIME_2,NUM_INFO_AFTER
238.
239.     ;加
240.     CALL DO_ADD
241.
242.     ;减
243.     CALL DO_SUB
244.
245.     ;乘
246.     CALL DO_MUL
247.
248.     ;除
249.     CALL DO_DIV
250.
251.     ;反复执行程序
252.     DO_AGAIN
253.
254.
255. ;---------------------------------------PROC
256.
257. ;加法
258. DO_ADD PROC
259.
260.     MOV AX,MTIME_1
261.     MOV BX,MTIME_2
262.     ADD AX,BX
263.     ;打印相加结果
264.     PRINTS ADD_INFO
265.     PRINT_NUMS AX
266.     NEXTLINE
```

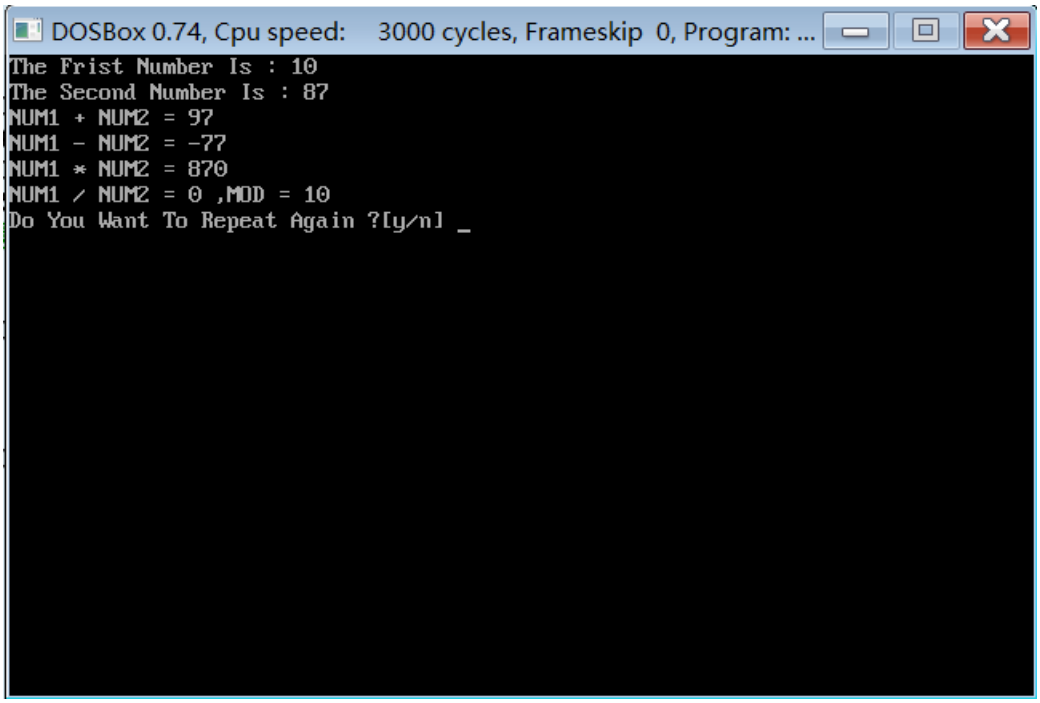
```
    RET

DO_ADD ENDP


;减法
DO_SUB PROC

    MOV AX,MTIME_1
    MOV BX,MTIME_2
    CMP AX,BX
    JAE SUB_NN
    ;如果是负数先打印‘-’
    XCHG AX,BX
    SUB AX,BX
    PRINTS SUB_INFO
    PRINTC '-'
    PRINT_NUMS AX
    NEXTLINE
    RET
    ;是正数直接打印
   SUB_NN:
    SUB AX,BX
    PRINTS SUB_INFO
    PRINT_NUMS AX
    NEXTLINE
    RET

DO_SUB ENDP

;乘法
DO_MUL PROC
    MOV AX,MTIME_1
    MOV BX,MTIME_2
    MUL BX
    ;打印相乘结果
    PRINTS MUL_INFO
    PRINT_NUMS AX
    NEXTLINE
    RET
DO_MUL ENDP

;除法
DO_DIV PROC
```

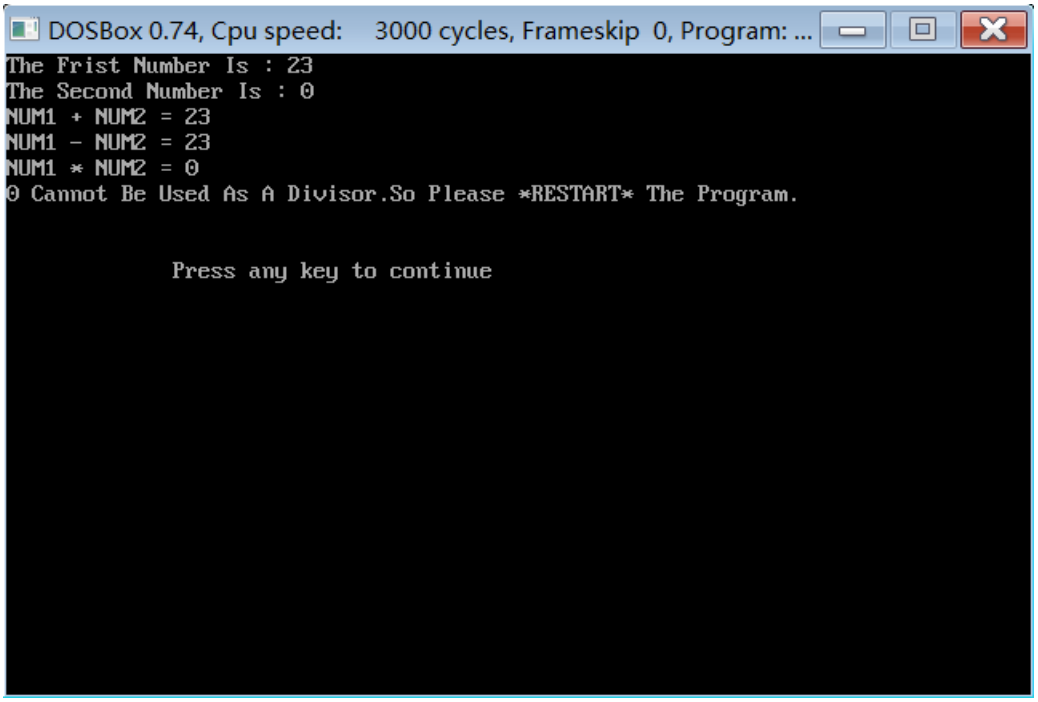
```
311.
312.     MOV AX,MTIME_1
313.     MOV BX,MTIME_2
314.     DIV BX
315.     ;打印商
316.     PRINTS DIV_INFO
317.     PRINT_NUMS AX
318.     ;打印余数
319.     PRINTS MOD_INFO
320.     PRINT_NUMS DX
321.     NEXTLINE
322.     RET
323.
324. DO_DIV ENDP
325.
326. ;延时
327. DELAY PROC
328.
329.     ;读取时钟
330.     MOV AH,00H
331.     INT 1AH
332.     MOV BX,DX
333.     ;反复查询时间差额是否满足
334.    REPTS:
335.     MOV AH,00H
336.     INT 1AH
337.     MOV AX,DX
338.     SUB AX,BX
339.     ;0s<AX<1s
340.     CMP AX,14
341.     JL REPTS
342.     CMP DELAY_TIME,30
343.     JBE DO_REPE_ADD
344.     JA DO_REPE_SUB
345.    ;延时时间反复游弋（8~32）
346.    DO_REPE_ADD:
347.     MOV BL,2
348.     ADD DELAY_TIME,BL
349.     RET
350.    DO_REPE_SUB:
351.     MOV BL,24
352.     SUB DELAY_TIME,BL
353.     RET
354.
```

```
355. DELAY ENDP
356.
357. CODES ENDS
358.     END START
```

运行截图：






DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
The Frist Number Is : 23
The Second Number Is : 0
NUM1 + NUM2 = 23
NUM1 - NUM2 = 23
NUM1 * NUM2 = 0
0 Cannot Be Used As A Divisor.So Please *RESTART* The Program.
Press any key to continue