

2152118 史君宝 汇编语言 大作业报告

一、设计题目：

题目原题：

某学院举行联欢晚会，在入口签到处登记参会学生的学号存入 `stu.txt` 文件（文件为只含 7 位数学号的文本文件，请自行用记事本产生该文件，也可直接使用附件中的 `mystu.txt` 文件。编一开奖程序，打开并读取文件中的数据，计算参会人数 N ，利用 BIOS 的 1AH 时钟中断产生一随机数 M ($1 \sim N$ 之间)，根据该随机数确定本次中奖人员为文件中的第 M 个学生，并在屏幕上显示该幸运数和中奖学号。可适当考虑显示的效果和特色。

题目解读：

由上面的题意可以知晓，我们需要完成很多的操作，总结来说我们需要将内容分为打开文件后进行读取，并对获得的随机数进行处理，输出最终获奖学生的结果。为了进行美化处理，我们需要对最终的输出结果进行一定的处理。

二、设计说明

1. 程序功能：

- (1) 打开文件，然后读取文件，最后要关闭文件。
- (2) 在读取文件的过程中，将读到的所有的学生学号输出出来，每 8 个为一行。并对所有读取到的学生进行计数。
- (3) 设计一个求随机数的函数，并根据结果求出对应的幸运学生的学号。
- (4) 完成程序的美观和优化。

2. 子程序及各部分之间的关系

(1) 程序的数据：

ADDRASS	用于储存的文件名称和位置
BUFFER	作为输入数据缓冲区，帮助我们在读取文件的过程中存储数据
STR_open	未成功打开文件的提示字段
STR_read	未成功读取文件的提示字段
STR_close	未成功关闭文件的提示字段
STR_stu	将所有的学生列举出来的提示字段
STR_num	学生总数的提示字段
STR_luck	幸运学生在整个名单的第几名
STR_order	幸运学生的学号为

FLAG	标志位，帮助我们判断在打开读取和关闭文件的过程是否正确
sum	变量，用于储存最终学生名单中的总数
ID	变量，用于记录学号的已输出位数，帮助成功输出 7 位学号
Line	变量，记录每行的学号的已输出个数，帮助成功在一行输出 8 个学号
Luck	变量，用于储存幸运学生在名单中的位置

8	DATA	SEGMENT		
9				
10	ADDRESS	DB	'C:\MYSTU.TXT',0	;储存的文件名称和位置
11				
12	BUFFER	DB	900 DUP (?)	;输入数据缓冲区
13				
14	STR_open	DB	13,10,'Fail to open the file! ',13,10,'\$'	;未成功打开文件的提示
15				
16	STR_read	DB	13,10,'Fail to read the file! ',13,10,'\$'	;未成功读取文件的提示
17				
18	STR_close	DB	13,10,'Fail to close the file! ',13,10,'\$'	;未成功关闭文件的提示
19				
20	STR_stu	DB	13,10,'All students are following: ',13,10,'\$'	;所有的学生列举
21				
22	STR_num	DB	13,10,'The number of students is ',13,10,'\$'	;学生总数为
23				
24	STR_luck	DB	'. ',13,10,'The lucky student is ',13,10,'\$'	;幸运的学生号码为
25				
26	STR_order	DB	13,10,'His number is ',13,10,'\$'	;幸运学生号码为
27				
28	FLAG	DB	0	;标志位
29				
30	sum	DB	0	
31	ID	DB	0	
32	Line	DB	0	
33	Luck	DB	0	
34	DATA	ENDS		
35				

(2) 主程序

首先是数据段的导入，然后进入 OPEN 函数，在其中实现文件的打开，读取和关闭，并在过程中输出学生名单，并记录学生总数。

然后调用 OUT2 函数，将学生总数输出出来。

然后调用 RANDD 函数，根据对应的中断时钟生成一个随机数。

然后调用 OUT3 和 OUT4 分别输出学生序号和学生学号。

43	START:		
44	MOV	AX,DATA	
45	MOV	DS,AX	
46	MOV	AX,0	
47			
48	CALL	OPEN	;调用OPEN函数读取文件
49			
50	CMP	FLAG,1	;如果文件操作错误，直接退出程序
51	JE	OVER	
52			
53	CALL	OUT2	;调用OUT2函数，输出第二句提示语与学生总人数N
54			
55	CALL	RANDD	;调用RANDD函数，生成随机数LUCK
56			
57	CALL	OUT3	;调用OUT3函数，输出第三句提示语与幸运学生序号M
58			
59	CALL	OUT4	;调用OUT4函数，输出幸运学生学号
60			
61	OVER:		
62	MOV	AX,4C00H	
63	INT	21H	
64			

(3) OPEN 函数:

首先看是不是能打开文件, AH = 3DH AL = 2 int 21H

如果 CF 为 1, 那么说明打不开文件, 转到错误处理 Error_open

然后看是不是能读取文件, AH = 3FH AL = 2 int 21H

如果 CF 为 1, 那么说明读取不了文件, 转到错误处理 Error_read

然后看是不是能关闭文件, AH = 3EH AL = 2 int 21H

如果 CF 为 1, 那么说明关闭不了文件, 转到错误处理 Error_close

在成功打开文件之后, 我们可以转到 OUT1 函数, 输出提示字符串
然后转到 COUT 函数, 用于输出所有的学生名单, 并完成计数。

```
05
66 OPEN PROC NEAR ;OPEN函数来操作文件
67 PUSH SI
68 PUSH AX
69 PUSH DX
70 PUSH CX
71 PUSH BX
72 MOV AH, 3DH ;打开文件
73 LEA DX, ADDRASS
74 MOV AL, 2
75 INT 21H
76 JC ERROR_open
77
78 LEA DX, BUFFER ;读取文件并将文件内容
79 MOV BX, AX
80 MOV CX, 900
81 MOV AH, 3FH
82 INT 21H
83 JC ERROR_read
84
85 CALL OUT1 ;调用OUT1函数, 输出第一句提示语
86 CALL COUT ;调用COUT函数, 显示全部学号并计数
87
88 MOV AH, 3EH ;关闭文件
89 INT 21H
90 JC ERROR_close
91 JMP return_OPEN
92
```

```
94
95 ERROR_open: ;文件未能成功打开反馈
96 MOV AH, 9
97 MOV FLAG, 1
98 LEA DX, STR_open
99 INT 21H
100 JMP return_OPEN
101 ERROR_read: ;文件未能成功读取反馈
102 MOV AH, 9
103 MOV FLAG, 1
104 LEA DX, STR_read
105 INT 21H
106 JMP return_OPEN
107 ERROR_close: ;文件未能成功关闭反馈
108 MOV AH, 9
109 MOV FLAG, 1
110 LEA DX, STR_close
111 INT 21H
112 JMP return_OPEN
113
114 return_OPEN:
115 POP BX
116 POP CX
117 POP DX
118 POP AX
119 POP SI
120 RET
121
122 OPEN ENDP
123
```

(4) OUT1 函数

使 AH = 09H AL = 2 int 21H 实现提示字符串的输出
字符串的样式:

```
STR_stu      DB    13,10,'All students are following:',13,10,'$';所有的学生列举
```

(5) COUT 函数: 实现所有学生名单的输出, 并实现学生个数的计量
首先给 Line 赋值为 0, 用它来计量每一行的

```
136
137 COUT PROC NEAR ;COUT函数用于将文件内容打印在显示屏上并计数
138     PUSH AX
139     PUSH CX
140     PUSH DX
141     PUSH SI
142     MOV Line, 0
143     MOV CX, AX
144     LEA SI, BUFFER
145     MOV AH, 2
146 C1:
147     MOV DH,[SI]
148     CMP Line, 8 ;每行显示8个学号, 显示完毕转行
149     JNE C2
150     MOV Line, 0
151     MOV DL, 13
152     INT 21H
153     MOV DL, 10
154     INT 21H
155 C2:
156     CMP DH, 10
157     JNE C3
158     MOV DL, 32
159     INT 21H
160     INC Line
161     JMP C4
162 C3:
163     CMP DH, 13
164     JE C4
165     MOV DL, DH
166     INT 21H
167 C4:
168     CALL COUNT ;调用COUNT函数, 检查[SI]并计数
169     INC SI
170     LOOP C1
171
172     POP SI
173     POP DX
174     POP CX
175     POP AX
176     RET
177 COUT ENDP
```

(6) Print_ten 函数 实现将十六进制数转换为十进制数输出
首先判断是否大于 10:
小于 10 转到一位数据处理程序, 否则转到三位数据处理程序, 进一步看是否是一个三位数。

```

182 Print_ten PROC NEAR
183     PUSH    AX
184     PUSH    BX
185     PUSH    CX
186     PUSH    DX
187     CMP     sum, 10      ;判断N是否只有一位
188     JS      P1
189     JMP     P3
190

```

一位数的处理程序：

将数据加上 30H 之后，使得 AH = 02H，用 int 21H 就可以实现十进制输出。

```

91 P1:
92     ADD     sum, 30H
93     MOV     AH, 2
94     MOV     DL, sum
95     INT     21H
96     JMP     return_P
97

```

三位数的处理程序：

首先将原来的除以 100，判断商是否为 0，如果为 0，转到两位数的处理程序，否则继续留在三位数处理程序。

```

197
198 P3:
199     MOV     AL, sum      ;AX 作为被除数，扩展到16位
200     MOV     AH, 0
201     MOV     BL, 100     ;BL 作为除数，为8位，是100
202     DIV     BL          ;AL 是商，AH 是余数
203
204     CMP     AL, 0
205     JZ      P2
206
207     MOV     BH, AH
208     MOV     DL, AL
209     ADD     DL, 30H
210     MOV     AH, 02H
211     INT     21H
212
213     MOV     AL, BH
214     MOV     AH, 0
215
216     MOV     BL, 10
217     DIV     BL
218     MOV     BL, AH
219     MOV     DL, AL
220     ADD     DL, 30H
221     MOV     AH, 02H
222     INT     21H
223     MOV     DL, BL
224     ADD     DL, 30H
225     INT     21H
226     JMP     return_P
227

```


两位数处理程序：

```
227
228 P2:
229     MOV     AL, AH
230     AND     AH, 0
231     MOV     BL, 10
232     DIV     BL
233     MOV     BL, AH
234     MOV     DL, AL
235     ADD     DL, 30H
236     MOV     AH, 02H
237     INT     21H
238     MOV     DL, BL
239     ADD     DL, 30H
240     INT     21H
241     JMP     return_P
242
```

(7) COUNT 函数

首先我们将所有的学号的每一个数位分别进行比较，使其在 30H 到 39 之间，使 ID 加 1，当 ID = 7 的时候，就能够使得 sum + 1

```
253
254 COUNT PROC NEAR           ;COUNT函数用于记录总人数
255     PUSH    DX
256     CMP     DH, 30H
257     JB      AB
258     CMP     DH, 39H
259     JA      AB
260     INC     ID
261
262 AB:
263     CMP     ID, 7
264     JNE     return_C
265     INC     sum
266     MOV     ID, 0
267     JMP     return_C
268
269 return_C:
270     POP     DX
271     RET
272
273 COUNT ENDP
274
```

(8) OUT2 函数

使 AH = 09H AL = 2 int 21H 实现提示字符串的输出
字符串的样式：

```
1
2 STR_num DB 13,10,'The number of students is',13,10,'$' ;学生总数为
3
```

然后调用 Print_ten 程序，将总人数输出

```

:76
:77 OUT2 PROC NEAR ;OUT2用于输出第二句提示字符串
:78     PUSH AX
:79     PUSH DX
:80     MOV AH, 09H
:81     LEA DX, STR_num
:82     INT 21H
:83     CALL Print_ten ;调用Print_ten函数输出与会总人数
:84     POP DX
:85     POP AX
:86     RET
:87 OUT2 ENDP

```

(9) OUT3 函数

使 AH = 09H AL = 2 int 21H 实现提示字符串的输出
字符串的样式：

```
STR_luck DB '.',13,10,'The lucky student is',13,10,'$' ;幸运的学生号码为
```

然后将幸运号码赋值给 sum，进一步通过 Print_ten 来实现十进制输出。

```

:91 OUT3 PROC NEAR ;OUT3用于输出第三句提示字符串
:92     PUSH AX
:93     PUSH BX
:94     PUSH DX
:95     MOV DH, sum
:96     MOV AH, 09H
:97     LEA DX, STR_luck
:98     INT 21H
:99     MOV BL, Luck
:00     MOV sum, BL
:01     CALL Print_ten ;调用PRINN函数，将M以十进制输出
:02     MOV AH, 09H
:03     LEA DX, STR_order
:04     INT 21H
:05     MOV sum, DH
:06     POP DX
:07     POP BX
:08     POP AX
:09     RET
:10 OUT3 ENDP

```

(10) OUT4 函数

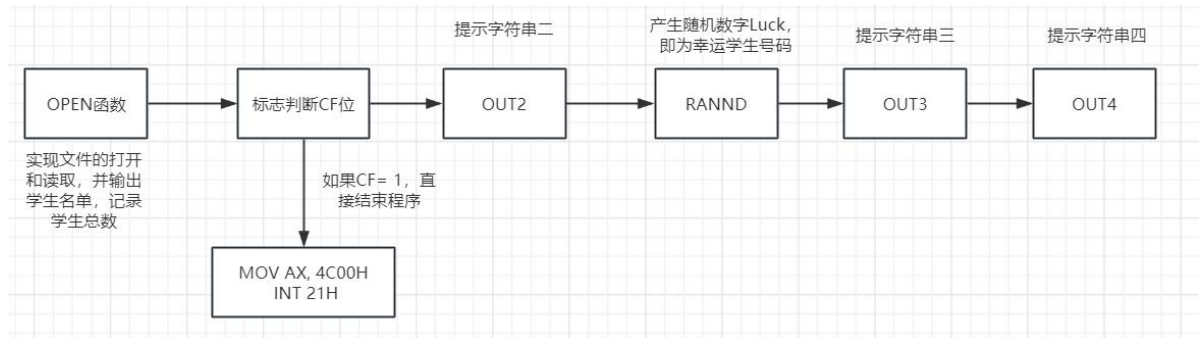
首先将中奖号码的位置求出来，根据每个有 7 位学号和一部份的后缀，我们可以知道真正的中奖号码的位置，然后我们使得 CX = 7，通过逐个输出字符我们就可以将最终的学号输出。

提示字符串：

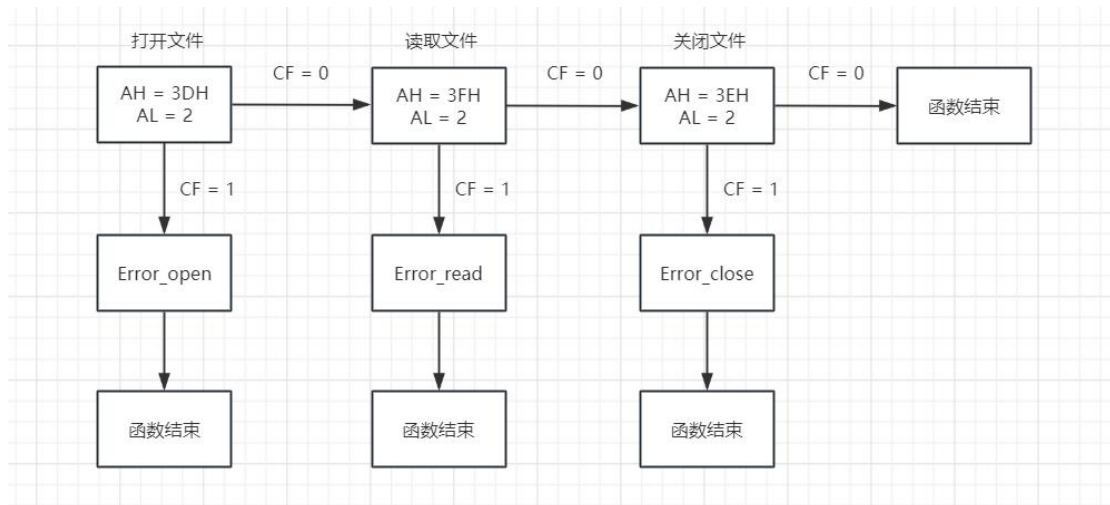
```
STR_order DB 13,10,'His number is',13,10,'$' ;幸运学生号码为
```

3. 程序框图

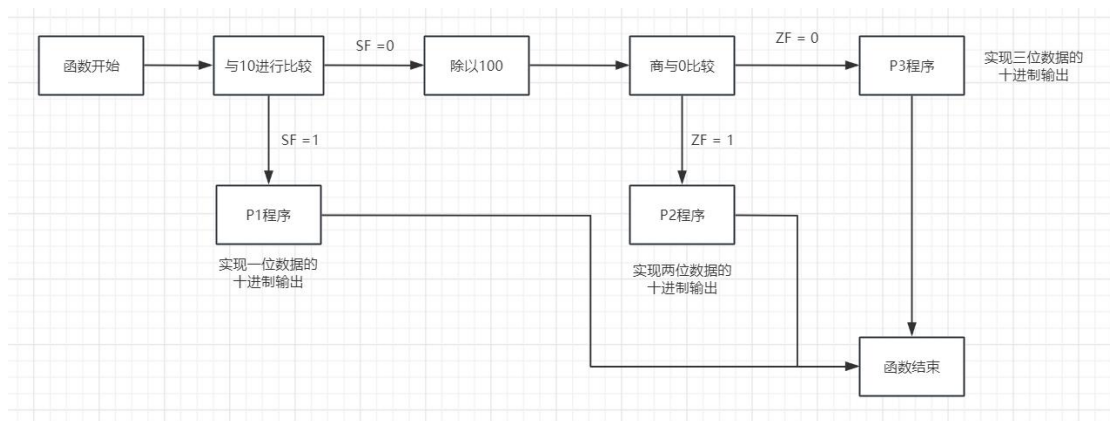
(1) 主函数的程序框图



(2) OPEN 函数程序框图



(3) Print_ten 函数程序框图



三、使用说明

使用方法：

我们将文中所需要的 mystu.txt 文件导入我们的 C 盘中，然后利用 DOSXBOX 就能够使用，并输出最终结果。

结果显示：

```
C:\>second.exe

All students are following:
2050123 1950222 2151439 1953010 2054571 1956012 2056098 2152023
1953768 2051035 2154457 1952056 1952139 1953123 2053567 2151561
2154028 2053098 2055567 1951045 1950013 2105789 1952021 1951032
2152134 1952018 2154468 2151045 2054321 2153290 1956311 1954064
2152534 2150123 1955367 2105200 2051234 2150123
The number of students is
38.
The lucky student is
8
His number is
2152023
```

```
C:\>second.exe

All students are following:
2050123 1950222 2151439 1953010 2054571 1956012 2056098 2152023
1953768 2051035 2154457 1952056 1952139 1953123 2053567 2151561
2154028 2053098 2055567 1951045 1950013 2105789 1952021 1951032
2152134 1952018 2154468 2151045 2054321 2153290 1956311 1954064
2152534 2150123 1955367 2105200 2051234 2150123
The number of students is
38.
The lucky student is
16
His number is
2151561
```

```
C:\>second.exe

All students are following:
2050123 1950222 2151439 1953010 2054571 1956012 2056098 2152023
1953768 2051035 2154457 1952056 1952139 1953123 2053567 2151561
2154028 2053098 2055567 1951045 1950013 2105789 1952021 1951032
2152134 1952018 2154468 2151045 2054321 2153290 1956311 1954064
2152534 2150123 1955367 2105200 2051234 2150123
The number of students is
38.
The lucky student is
36
His number is
2105200
```

```
C:\>second.exe

All students are following:
2050123 1950222 2151439 1953010 2054571 1956012 2056098 2152023
1953768 2051035 2154457 1952056 1952139 1953123 2053567 2151561
2154028 2053098 2055567 1951045 1950013 2105789 1952021 1951032
2152134 1952018 2154468 2151045 2054321 2153290 1956311 1954064
2152534 2150123 1955367 2105200 2051234 2150123
The number of students is
38.
The lucky student is
17
His number is
2154028
```

四、课程总结

1. 大作业总结

(1) 问题总结，通过之前的学习，本次大作业我选择的是最后一题，是实现对文件的打开和读取，并通过随机数函数产生随机数，然后从中挑选出最终获奖的学生序号和对应的学号。比较综合考察文件的读取和字符字符串的输入输出等知识，比较困难。

(2) 难点总结：在这次大作业中文件的操作就十分困难了，因为打开读取和关闭的 AH 值各不同，知识点复杂难记，在最开始读取的时候就遇到了问题。

然后对于程序中用到的大量的学号信息的储存，信息量较大，同时其中包含空格和回车，对于这些的读取就比较困难了。

最后就是对应的字符和字符串的输出，以及将数据转换为 16 进制输出出来，都是比较困难的，因为这不像是 C++ 等语言中能够比较轻松。

2. 课程总结

(1) 课程掌握情况：整体学了下来，我可以清楚地感受到内容是十分多的，仅仅 10 天的课程，在前两天入门和考试去掉，内容较多，但是 PPT 就已经达到六七百面，而且其中的内容也缺乏规律性，很难清楚地记忆下来。

在完成作业的过程中，就能够明显感受到，首先知识太底层了，想要完成原来程序语言的一个小任务，就需要大量的汇编语句，而且还有很多的标号和跳转，总的来说知识不熟，总是感觉写代码时缺乏称手的工具，很容易卡住。

(2) 课程建议：课程内容太多，这不是问题，上课解决不了也不是问题，但是 PPT 要精美呀，我已经自学看了 PPT 三四遍了，但是有些东西还是不懂，有的东西 PPT 上涉及了，但是讲的一知半解，非常难受。现在脑袋里还是一团浆糊，建议再丰富一下 PPT，增加一些例子和内存单元的显示。

五、程序原码

```
6
7 DATA SEGMENT
8
9 ADDRESS DB 'C:\MYSTU.TXT',0 ;储存的文件名称和位置
10
11 BUFFER DB 900 DUP (?) ;输入数据缓冲区
12
13 STR_open DB 13,10,'Fail to open the file!',13,10,'$' ;未成功打开文件的提示
14
15 STR_read DB 13,10,'Fail to read the file!',13,10,'$' ;未成功读取文件的提示
16
17 STR_close DB 13,10,'Fail to close the file!',13,10,'$' ;未成功关闭文件的提示
18
19 STR_stu DB 13,10,'All students are following:',13,10,'$' ;所有的学生列举
20
21 STR_num DB 13,10,'The number of students is',13,10,'$' ;学生总数为
22
23 STR_luck DB '.',13,10,'The lucky student is',13,10,'$' ;幸运的学生号码为
24
25 STR_order DB 13,10,'His number is',13,10,'$' ;幸运学生号码为
26
27 FLAG DB 0 ;标志位
28
29 sum DB 0
30 ID DB 0
31 Line DB 0
32 Luck DB 0
33 DATA ENDS
```

```

36 SSEG SEGMENT 'STACK'
37 DW 200H DUP(0)
38 SSEG ENDS
39
40 CODE SEGMENT
41 ASSUME CS:CODE,DS:DATA,SS:SSEG
42
43 START:
44     MOV     AX,DATA
45     MOV     DS,AX
46     MOV     AX,0
47
48     CALL    OPEN           ;调用OPEN函数读取文件
49
50     CMP     FLAG,1         ;如果文件操作错误，直接退出程序
51     JE      OVER
52
53     CALL    OUT2           ;调用OUT2函数，输出第二句提示语与学生总人数N
54
55     CALL    RANDD          ;调用RANDD函数，生成随机数LUCK
56
57     CALL    OUT3           ;调用OUT3函数，输出第三句提示语与幸运学生序号M
58
59     CALL    OUT4           ;调用OUT4函数，输出幸运学生学号
60
61 OVER:
62     MOV     AX, 4C00H
63     INT     21H

```

```

65
66 OPEN PROC NEAR           ;OPEN函数来操作文件
67     PUSH    SI
68     PUSH    AX
69     PUSH    DX
70     PUSH    CX
71     PUSH    BX
72     MOV     AH, 3DH        ;打开文件
73     LEA     DX, ADDRASS
74     MOV     AL, 2
75     INT     21H
76     JC      ERROR_open
77
78     LEA     DX, BUFFER     ;读取文件并将文件内容
79     MOV     BX, AX
80     MOV     CX, 900
81     MOV     AH, 3FH
82     INT     21H
83     JC      ERROR_read
84
85     CALL    OUT1           ;调用OUT1函数，输出第一句提示语
86     CALL    COUT           ;调用COUT函数，显示全部学号并计数
87
88     MOV     AH,3EH        ;关闭文件
89     INT     21H
90     JC      ERROR_close
91     JMP     return_OPEN
92

```

```

94
95 ERROR_open:
96     MOV     AH,9           ;文件未能成功打开反馈
97     MOV     FLAG,1
98     LEA     DX,STR_open
99     INT     21H
100    JMP     return_OPEN
101 ERROR_read:
102     MOV     AH,9           ;文件未能成功读取反馈
103     MOV     FLAG,1
104     LEA     DX,STR_read
105     INT     21H
106     JMP     return_OPEN
107 ERROR_close:
108     MOV     AH,9           ;文件未能成功关闭反馈
109     MOV     FLAG,1
110     LEA     DX,STR_close
111     INT     21H
112     JMP     return_OPEN
113
114 return_OPEN:
115     POP     BX
116     POP     CX
117     POP     DX
118     POP     AX
119     POP     SI
120     RET
121
122 OPEN     ENDP

```

```

124
125 OUT1     PROC     NEAR           ;OUT1用于输出第一句提示字符串，表示接下来要输出所有的学生
126     PUSH     AX
127     PUSH     DX
128     MOV     AH, 09H
129     LEA     DX, STR_stu
130     INT     21H
131     POP     DX
132     POP     AX
133     RET
134 OUT1     ENDP

```

```

136
137 COUT     PROC     NEAR           ;COUT函数用于将文件内容打印在显示屏上并计数
138     PUSH     AX
139     PUSH     CX
140     PUSH     DX
141     PUSH     SI
142     MOV     Line, 0
143     MOV     CX, AX
144     LEA     SI, BUFFER
145     MOV     AH, 2
146 C1:
147     MOV     DH,[SI]
148     CMP     Line, 8           ;每行显示8个学号，显示完毕转行
149     JNE     C2
150     MOV     Line, 0
151     MOV     DL, 13
152     INT     21H
153     MOV     DL, 10
154     INT     21H
155 C2:
156     CMP     DH, 10
157     JNE     C3
158     MOV     DL, 32
159     INT     21H
160     INC     Line
161     JMP     C4
162 C3:
163     CMP     DH, 13
164     JE      C4
165     MOV     DL, DH
166     INT     21H

```

```

167 C4:
168     CALL    COUNT    ;调用COUNT函数，检查[SI]并计数
169     INC     SI
170     LOOP    C1
171
172     POP     SI
173     POP     DX
174     POP     CX
175     POP     AX
176     RET
177 COUT    ENDP
178

```

```

182 Print_ten    PROC    NEAR
183     PUSH    AX
184     PUSH    BX
185     PUSH    CX
186     PUSH    DX
187     CMP     sum, 10    ;判断N是否只有一位
188     JS      P1
189     JMP     P3
190
191 P1:
192     ADD     sum, 30H
193     MOV     AH, 2
194     MOV     DL, sum
195     INT     21H
196     JMP     return_P
197

```



```

198 P3:
199     MOV     AL, sum           ;AX 作为被除数, 扩展到16位
200     MOV     AH, 0
201     MOV     BL, 100          ;BL 作为除数, 为8位, 是100
202     DIV     BL               ;AL 是商, AH 是余数
203
204     CMP     AL, 0
205     JZ      P2
206
207     MOV     BH, AH
208     MOV     DL, AL
209     ADD     DL, 30H
210     MOV     AH, 02H
211     INT     21H
212
213     MOV     AL, BH
214     MOV     AH, 0
215
216     MOV     BL, 10
217     DIV     BL
218     MOV     BL, AH
219     MOV     DL, AL
220     ADD     DL, 30H
221     MOV     AH, 02H
222     INT     21H
223     MOV     DL, BL
224     ADD     DL, 30H
225     INT     21H
226     JMP     return_P
227

```

```

227
228 P2:
229     MOV     AL, AH
230     AND     AH, 0
231     MOV     BL, 10
232     DIV     BL
233     MOV     BL, AH
234     MOV     DL, AL
235     ADD     DL, 30H
236     MOV     AH, 02H
237     INT     21H
238     MOV     DL, BL
239     ADD     DL, 30H
240     INT     21H
241     JMP     return_P
242
243 return_P:
244     POP     DX
245     POP     CX
246     POP     BX
247     POP     AX
248     RET
249
250 Print_ten     ENDP
251

```

```

253
254 COUNT PROC NEAR ;COUNT函数用于记录总人数
255     PUSH    DX
256     CMP     DH, 30H
257     JB      AB
258     CMP     DH, 39H
259     JA      AB
260     INC     ID
261
262 AB:
263     CMP     ID, 7
264     JNE     return_C
265     INC     sum
266     MOV     ID, 0
267     JMP     return_C
268
269 return_C:
270     POP     DX
271     RET
272
273 COUNT ENDP
274

```

```

275
277 OUT2 PROC NEAR ;OUT2用于输出第二句提示字符串
278     PUSH    AX
279     PUSH    DX
280     MOV     AH, 09H
281     LEA     DX, STR_num
282     INT     21H
283     CALL    Print_ten ;调用Print_ten函数输出与会总人数
284     POP     DX
285     POP     AX
286     RET
287 OUT2 ENDP
288
289

```

```

290
291 OUT3 PROC NEAR ;OUT3用于输出第三句提示字符串
292     PUSH    AX
293     PUSH    BX
294     PUSH    DX
295     MOV     DH, sum
296     MOV     AH, 09H
297     LEA     DX, STR_luck
298     INT     21H
299     MOV     BL, Luck
300     MOV     sum, BL
301     CALL    Print_ten ;调用PRINN函数，将M以十进制输出
302     MOV     AH, 09H
303     LEA     DX, STR_order
304     INT     21H
305     MOV     sum, DH
306     POP     DX
307     POP     BX
308     POP     AX
309     RET
310 OUT3 ENDP

```

```

311
312 OUT4 PROC NEAR ;OUT4函数用于输出第三句提示语与幸运学生序号M
313     PUSH SI
314     PUSH AX
315     PUSH CX
316     PUSH DX
317     DEC Luck ;计算中奖学生学号在BUFFER数组中的位置
318     MOV AL, Luck
319     MOV DH, 9
320     MUL DH
321     MOV SI, AX
322     MOV CX, 7
323 HAHA:
324     MOV AH, 2 ;输出中奖学生的学号
325     MOV DL, BUFFER[SI]
326     INT 21H
327     INC SI
328     LOOP HAHA
329     POP DX
330     POP CX
331     POP AX
332     POP SI
333     RET
334 OUT4 ENDP
335

```

```

336
337 RANDD PROC NEAR ;RANDD函数用于生成随机数存入M
338     PUSH CX
339     PUSH DX
340     PUSH AX
341     STI
342     MOV AH, 0 ;读取时钟计数器的值
343     INT 1AH
344     MOV AX, DX
345     AND AH, 3
346     MOV DL, sum
347     DIV DL ;除以N, 取得余数再+1作为随机数
348     MOV BL, AH
349     INC BL
350     MOV Luck, BL
351     POP AX
352     POP DX
353     POP CX
354     RET
355 RANDD ENDP
356
357
358
359 CODE ENDS
360 END START

```