同济大学

汇编语言程序设计

（上机大作业报告）

**学 号 ： 1850000**

**姓 名 ： 一二三**

**专业/年级： 计算机类/大一**

**所在院系 ： 电信学院计算机系**

**任课教师 ： 朱程荣**

**完成日期 ： 2019年 0 月 00 日**

**一、设计题目：**

题目名称：文件操作

具体要求：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~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。

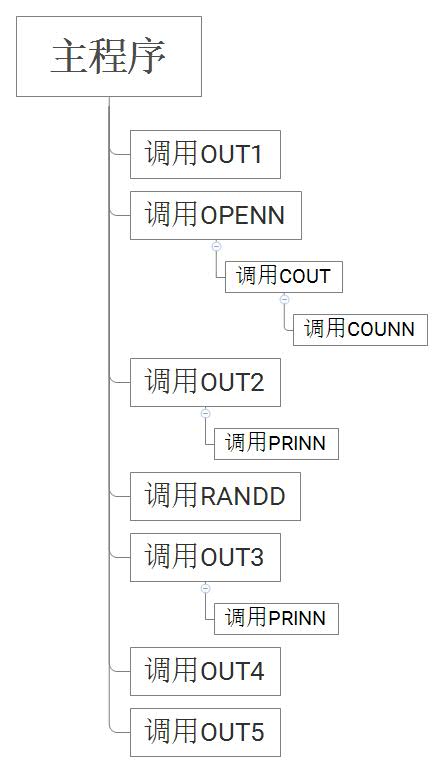
**二、设计说明：**

**2.1 程序功能综述**

本程序主要功能是读取存放在C:\位置下的MYSTU.TXT文件内容，并基于该文件中包含的7位数学号内容得出与会学生人数并显示。此外，本程序还将随机抽取一名到场学生为中奖学生，显示其学号存放的位次并显示其学号。

**2.2 任务划分**

本程序功能的实现主要分为①文件的打开、读取（OPENN）；②与会人数N的统计（COUNN）；③与会人数N的显示；④随机数M的生成；⑤随机数对应学号的显示。此外，为了使界面友好，任务还包括一些提示语的输出等。具体的子程序调用关系如下：



**2.3 子程序说明及其流程图**

以下，本文将依照子程序的调用先后，依次确切详细介绍每个子程序的功能、参数传递情况及寄存器占用情况，并给出该子程序的流程图：

·OUT1子程序：

·功能：输出第一句提示语“The all numbers are as follows:”

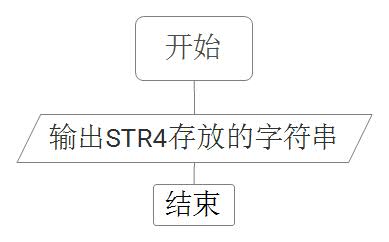
·入口参数：存储在DATA段STR4的字符串

·出口参数：无

·调用关系：由主程序直接调用

·占用寄存器：AX、DX

·流程图：



·OPENN子程序：

·功能：打开MYSTU.TXT文件，读取文件中的学号信息存入BUFFER，将读取到的学号信息输出在显示器上，同时统计与会人数存入N，待上述操作完毕后关闭文件。同时设置错误弹出机制，倘若文件在打开、读取、关闭过程中出现问题，立刻跳出程序并通过令FLAG=1提前结束进程

·入口参数：1）存储在DATA段STR1、STR2、STR3的字符串（依次输出提醒文件打开错误、文件读取错误及文件关闭错误）；

2）存储在DATA段ADDS的字符串（MYSTU.TXT存放地址）；

3）存储在DATA段BUFFER的暂存区域（用于储存从MYSTU.TXT读取的学号信息）；

4）存储在外部的MYSTU.TXT

·出口参数：1）读取完学号信息的BUFFER储存区域；

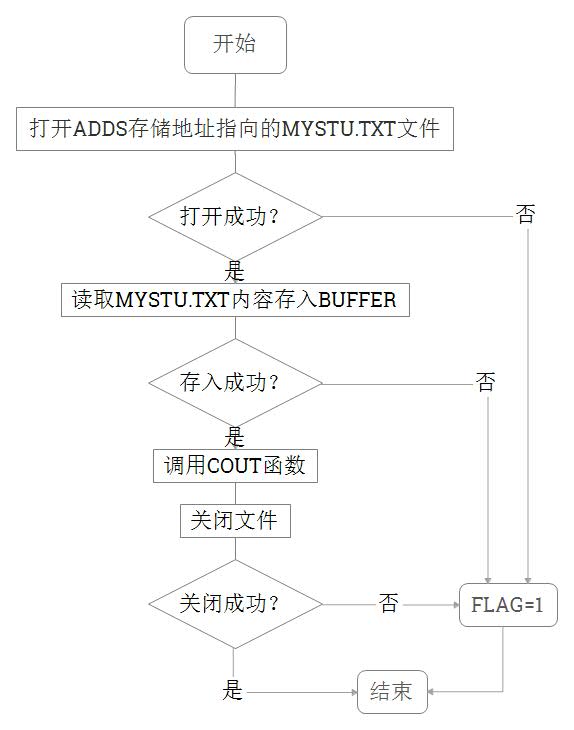
2）存储在DATA段的FLAG（用于判断文件操作是否顺利）；

3）存储在DATA段的N（用于储存与会人数）；

·调用关系：由主程序直接调用，该子程序内部调用COUT子程序

·占用寄存器：AX、BX、CX、DX、SI

·流程图：



·COUT子程序：

·功能：逐字节读取BUFFER中存储的内容并输出在显示器，为了美观保证每行显示七个学号，显示完毕再转行；同时对每一字节的内容进行判断，从而得到与会人数

·入口参数：存储完学号信息的BUFFER储存区域；

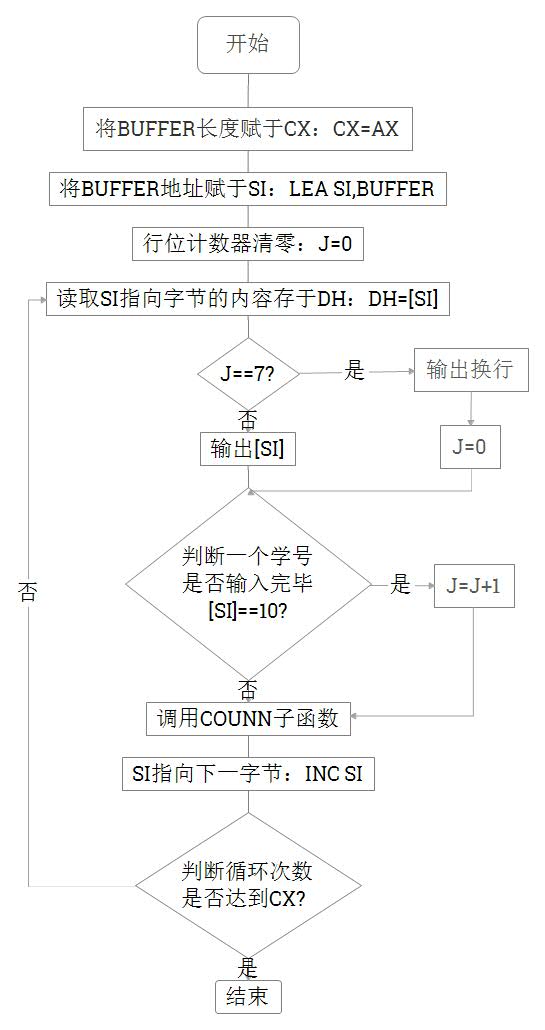
·出口参数：1）每次循环指向的待判断字节，存储于DH；

2）存储在DATA段的N（用于储存与会人数）；

·调用关系：由OPENN子程序调用，该子程序内部调用COUNN子程序

·占用寄存器：AX、CX、DX、SI

·流程图：



·COUNN子程序：

·功能：主要用于统计与会人数并存储于N

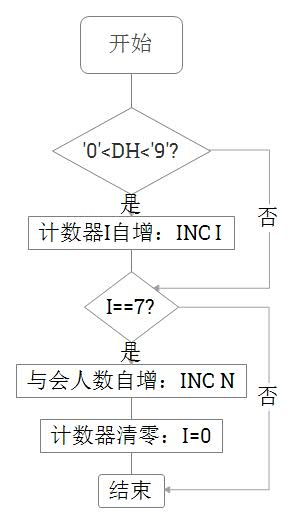
·入口参数：每次循环指向的待判断字节，存储于DH；

·出口参数：存储在DATA段的N（用于储存与会人数）；

·调用关系：由COUT子程序调用

·占用寄存器： DX

·流程图：



·OUT2子程序：

·功能：输出“The number of present students (means N) is ”，并调用PRINN函数将与会人数N按十进制形式输出

·入口参数：1）存储在DATA段STR5的字符串；

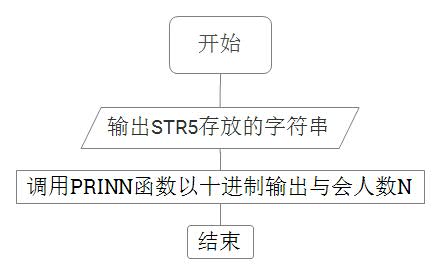
2）存储在DATA段的N（用于储存与会人数）；

·出口参数：无

·调用关系：由主程序直接调用

·占用寄存器： AX、DX

·流程图：



·PRINN子程序：

·功能：将N中存储的数据以十进制的方式输出于显示器

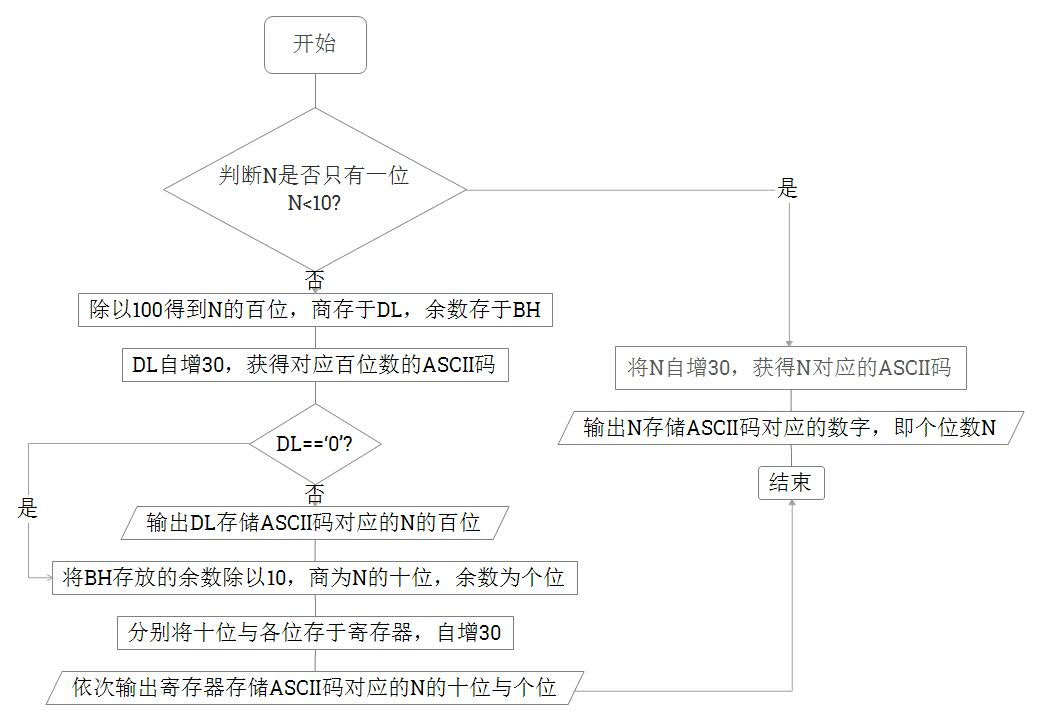
·入口参数：存储在DATA段的N（用于储存与会人数）；

·出口参数：无

·调用关系：由OUT2子程序调用，同时由OUT3子程序调用

·占用寄存器： AX、BX、CX、DX

·流程图：



·RANDD子程序：

·功能：生成一个介于[1,N]之间的随机数并存储于M，作为幸运学生的序号。

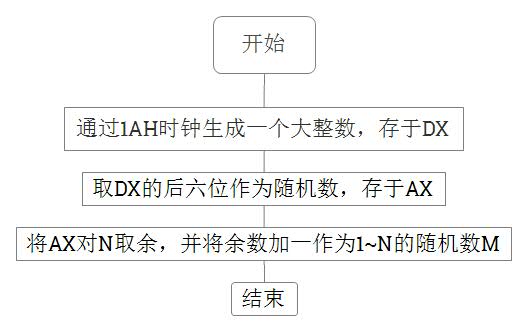
·入口参数：存储在DATA段的N（用于储存与会人数）；

·出口参数：存储在DATA段的M（用于储存幸运学生的序号）；

·调用关系：由主程序直接调用

·占用寄存器： AX、CX、DX

·流程图：



·OUT3子程序：

·功能：输出“.\nThe lucky student is the No.”，并调用PRINN函数将幸运学生序号M按十进制形式输出，再输出“one to be here.\nHis number is”

·入口参数：1）存储在DATA段STR6的字符串；

2）存储在DATA段的M（用于储存幸运学生的序号）；

·出口参数：无

·调用关系：由主程序直接调用

·占用寄存器： AX、BX、DX

·流程图：



·OUT4子程序：

·功能：根据M的值计算幸运学生学号在BUFFER中存储的序号，并输出该幸运学生的学号

·入口参数：1）存储在DATA段的M（用于储存幸运学生的序号）；

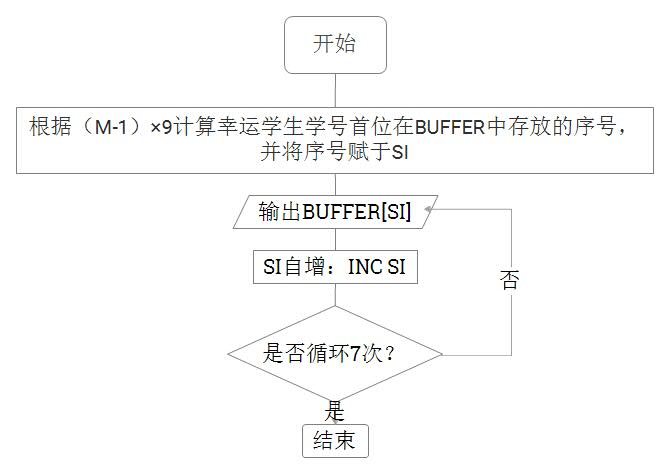
2）读取完学号信息的BUFFER储存区域；

·出口参数：无

·调用关系：由主程序直接调用

·占用寄存器： AX、CX、DX、SI

·流程图：



·OUT5子程序：

·功能：输出第四句提示语“\(^\_^)/ Congratulations!”

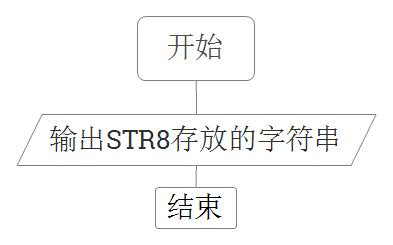
·入口参数：存储在DATA段STR8的字符串

·出口参数：无

·调用关系：由主程序直接调用

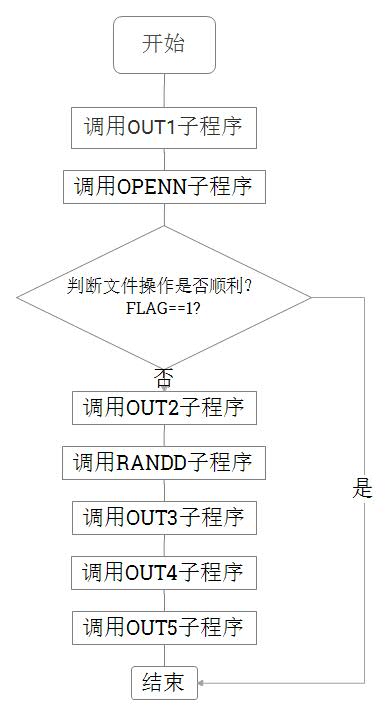
·占用寄存器：AX、DX

·流程图：



**2.4 程序框图**

主程序的流程框图如下：



**2.5 程序清单**

源程序所需的MYSTU.TXT文件存储在C:/目录下，该TXT文件即大要求附件中的mystu.txt文档，仅作重命名与转移操作。具体的程序清单如下：

;此处为定义存储段

DATA SEGMENT

ADDS DB 'C:\MYSTU.TXT',0

BUFFER DB 900 DUP (?)

STR1 DB 13,10,'Fail to open the file!$'

STR2 DB 13,10,'Fail to read the file!$'

STR3 DB 13,10,'Fail to close the file!$'

STR4 DB 13,10,'The all numbers are as follows:',13,10,'$'

STR5 DB 13,10,'The number of present students (means N) is $'

STR6 DB '.',13,10,'The lucky student is the No.$'

STR7 DB ' one to be here.',13,10,'His number is $'

STR8 DB '.',13,10,'\(^\_^)/ Congratulations!$'

FLAG DB 0

N DB 0

I DB 0

J DB 0

M DB 0

DATA ENDS

;此处为定义堆栈

SSEG SEGMENT 'STACK'

DW 200H DUP(0)

SSEG ENDS

;此处开始为代码段

CODE SEGMENT

ASSUME CS:CODE,DS:DATA,SS:SSEG

PRINN PROC NEAR ;PRINN函数用于将N以十进制形式输出

PUSH AX

PUSH BX

PUSH CX

PUSH DX

CMP N,10 ;判断N是否只有一位

JNB P2

ADD N,30H

MOV AH,2

MOV DL,N

INT 21H

JMP P3

P2: MOV AL,N

MOV BL,100

AND AX,00FFH

DIV BL

MOV BH,AH

MOV DL,AL

ADD DL,30H

CMP DL,30H

JZ P1

MOV AH,02H

INT 21H

P1: MOV BL,10

MOV AX,0

MOV AL,BH

DIV BL

MOV BH,AH

MOV DL,AL

ADD DL,30H

MOV AH,02H

INT 21H

MOV DL,BH

ADD DL,30H

INT 21H

P3: POP DX

POP CX

POP BX

POP AX

RET

PRINN ENDP

COUNN PROC NEAR ;COUNN函数用于记录总人数

PUSH DX

CMP DH,30H

JB ONE

CMP DH,39H

JA ONE

INC I

ONE: CMP I,7

JNE COEXIT

INC N

MOV I,0

COEXIT: POP DX

RET

COUNN ENDP

OUT1 PROC NEAR ;OUT1用于输出第一句提示字符串

PUSH AX

PUSH DX

MOV AH,9

LEA DX,STR4

INT 21H

POP DX

POP AX

RET

OUT1 ENDP

OUT2 PROC NEAR ;OUT2用于输出第二句提示字符串

PUSH AX

PUSH DX

MOV AH,9

LEA DX,STR5

INT 21H

CALL PRINN ;调用PRINN函数输出与会总人数

POP DX

POP AX

RET

OUT2 ENDP

COUT PROC NEAR ;COUT函数用于将文件内容打印在显示屏上并计数

PUSH AX

PUSH CX

PUSH DX

PUSH SI

MOV J,0

MOV CX,AX

LEA SI,BUFFER

MOV AH,2

C1: MOV DH,[SI]

CMP J,7 ;每行显示7个学号，显示完毕转行

JNE C2

MOV J,0

MOV DL,13

INT 21H

MOV DL,10

INT 21H

C2: CMP DH,10

JNE C3

MOV DL,32

INT 21H

INC J

JMP C4

C3: CMP DH,13

JE C4

MOV DL,DH

INT 21H

C4: CALL COUNN ;调用COUNN函数，检查[SI]并计数

INC SI

LOOP C1

POP SI

POP DX

POP CX

POP AX

RET

COUT ENDP

OPENN PROC NEAR ;OPENN函数用于操作文件

PUSH SI

PUSH AX

PUSH DX

PUSH CX

PUSH BX

MOV AH,3DH ;打开文件

LEA DX,ADDS

MOV AL,2

INT 21H

JC ERROR1

LEA DX,BUFFER ;读取文件并将文件内容

MOV BX,AX

MOV CX,900

MOV AH,3FH

INT 21H

JC ERROR2

CALL COUT ;调用COUT函数，显示全部学号并计数

MOV AH,3EH ;关闭文件

INT 21H

JC ERROR3

OPEXIT: POP BX

POP CX

POP DX

POP AX

POP SI

RET

ERROR1: MOV AH,9 ;文件未能成功打开反馈

MOV FLAG,1

LEA DX,STR1

INT 21H

JMP OPEXIT

ERROR2: MOV AH,9 ;文件未能成功读取反馈

MOV FLAG,1

LEA DX,STR2

INT 21H

JMP OPEXIT

ERROR3: MOV AH,9 ;文件未能成功关闭反馈

MOV FLAG,1

LEA DX,STR3

INT 21H

JMP OPEXIT

OPENN ENDP

RANDD PROC NEAR ;RANDD函数用于生成随机数存入M

PUSH CX

PUSH DX

PUSH AX

STI

MOV AH,0 ;读取时钟计数器的值

INT 1AH

MOV AX,DX

AND AH,3

MOV DL,N

DIV DL ;除以N，取得余数再+1作为随机数

MOV BL,AH

INC BL

MOV M,BL

POP AX

POP DX

POP CX

RET

RANDD ENDP

OUT3 PROC NEAR ;OUT3用于输出第三句提示字符串

PUSH AX

PUSH BX

PUSH DX

MOV DH,N

MOV AH,9

LEA DX,STR6

INT 21H

MOV BL,M

MOV N,BL

CALL PRINN ;调用PRINN函数，将M以十进制输出

MOV AH,9

LEA DX,STR7

INT 21H

MOV N,DH

POP DX

POP BX

POP AX

RET

OUT3 ENDP

OUT4 PROC NEAR ;OUT4函数用于输出第三句提示语与幸运学生序号M

PUSH SI

PUSH AX

PUSH CX

PUSH DX

DEC M ;计算中奖学生学号在BUFFER数组中的位置

MOV AL,M

MOV DH,9

MUL DH

MOV SI,AX

MOV CX,7

HAHA: MOV AH,2 ;输出中奖学生的学号

MOV DL,BUFFER[SI]

INT 21H

INC SI

LOOP HAHA

POP DX

POP CX

POP AX

POP SI

RET

OUT4 ENDP

OUT5 PROC NEAR ;OUT1用于输出第四句提示字符串

PUSH AX

PUSH DX

MOV AH,9

LEA DX,STR8

INT 21H

POP DX

POP AX

RET

OUT5 ENDP

START: MOV AX,DATA

MOV DS,AX

MOV AX,0

CALL OUT1 ;调用OUT1函数，输出第一句提示语

CALL OPENN ;调用OPENN函数对文件进行打开、读取、存储、关闭，同时

; 计算与会学生总人数N

CMP FLAG,1 ;如果文件操作错误，直接退出程序

JE OVER

CALL OUT2 ;调用OUT2函数，输出第二句提示语与学生总人数N

CALL RANDD ;调用RANDD函数，生成随机数LUCK

CALL OUT3 ;调用OUT3函数，输出第三句提示语与幸运学生序号M

CALL OUT4 ;调用OUT4函数，输出幸运学生学号

CALL OUT5 ;调用OUT5函数，输出第四句提示语

OVER: MOV AX,4C00H

INT 21H

CODE ENDS

END START

**三、调试说明：**

**3.1 调试情况**

**3.1.1 所遇问题及解决方法**

上机过程大体来说较为顺利，较大一部分所遇到的问题主要是在于错误拼写、中英文分号未区分等较为低级且易解决的方面。以下主要列举几点对我今后编程有所启发的错误：

1）不能很好地理解并掌握文件内指针的操作方式，致使无法有效地按照逐字节的方式对文件内储存学号进行读取、查找操作。

解决方案：

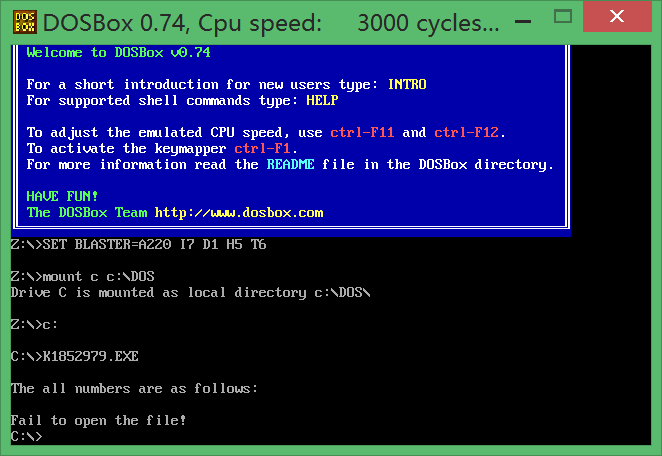
放弃通过调节文件内指针从而达到逐字节读取的方式，改为将文件存储内容全部放入BUFFER暂存区域，从而将不熟悉的文件内指针操作改换为熟悉的DATA段指针操作。

2）原本计划将M和N的十进制输出分别调用两个子程序完成，但在编写时发现这两个子程序的代码重复度很高。

解决方案：

仅编写十进制输出N的子程序PRINN，需要输出M时，将M的值暂存给N，再次调用PRINN子程序，从而提高代码的复用率。

3）刚开始编写时，发现程序无法顺利打开MYSTU.TXT文件，显示“Fail to open the file!”截屏如下：



解决方案：

多次调试无果后，意识到自己所使用的DOSBox会在程序编译过程中将DOS文件夹虚拟为C:\目录，所以在ADDS处赋于MYSTU.TXT真正的系统存放路径时会出现无法找寻的情况。将ADDS处的路径更换为“C:\MYSTU.TXT”后，程序得以正常运行。

**3.1.2 技巧分析与总结**

通过对上机过程中所遇到上述问题的思考与解决，我认为自己在日后的编程过程中应该做到以下几点：

1）不论何种编程语言与方式，都应在开始学习编程前了解清楚自己所使用的编译环境及开发工具的特性及使用方式，以免出现浪费大量时间找寻文件路径的这种情况再次发生。

2）当面对一个较大的程序工程时，应该在最开始设计程序模块的时候就考虑到代码复用的问题，力求使自己的程序能做到多快好省。

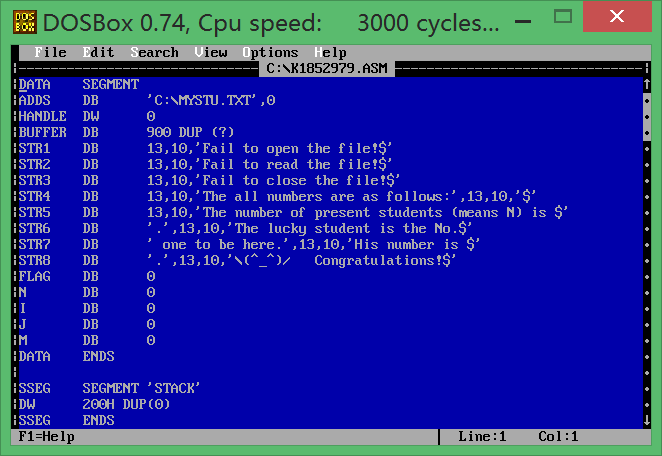
3）今后学习编程知识时，面对未接触过的工具与要求，不要一昧地死磕死学知识点，应该同时也考虑将未学知识点转化为自己熟悉的知识点的可能性。

**3.2 连接情况**

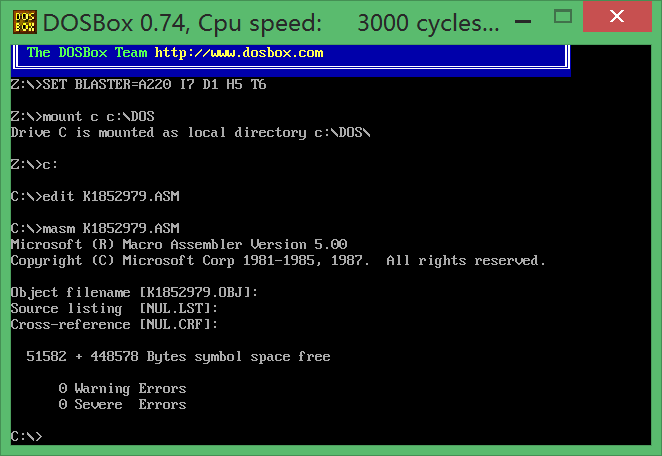
考虑到本程序仅含有一个ASM文件，所以并无特殊的连接要求，只需将K1850000.ASM汇编而成的K1850000.OBJ文件连接为.EXE文件即可。连接流程示意图如下：

（该图片设计学号信息，删去）

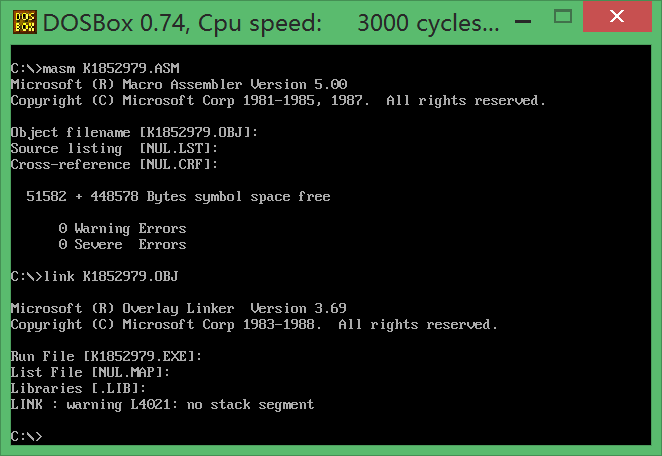
首先在edit界面对K1850000.ASM进行编辑，编辑截图如下：



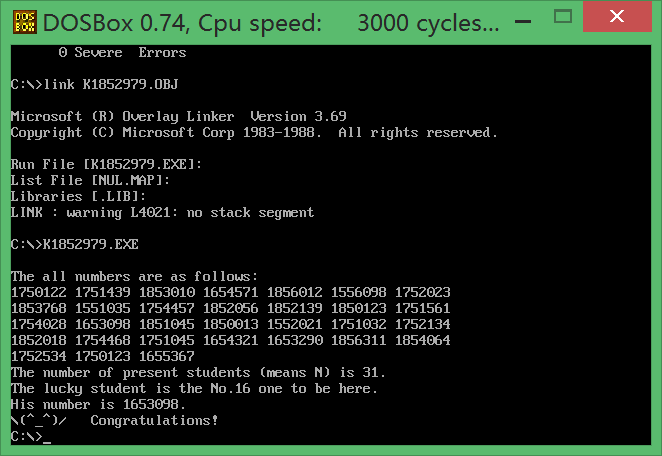
编辑结束后，对K1850000.ASM进行汇编，汇编截图如下：



汇编结束后，对K1850000.OBJ进行连接，连接截图如下：



连接结束后，运行K1850000.EXE，运行截图如下：



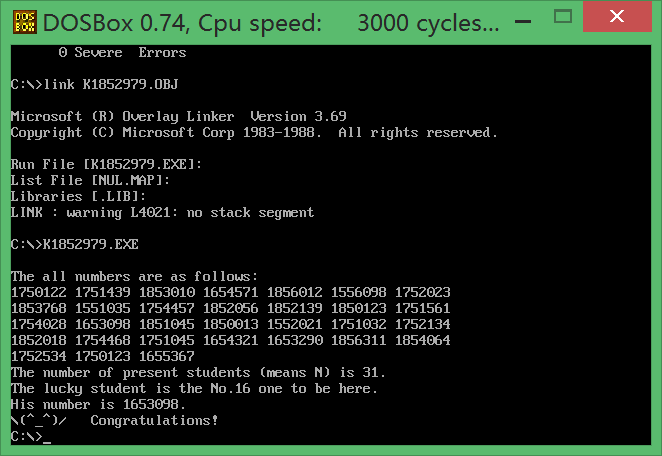
运行结果与预期该程序要求达到的功能相符合，且未出现宕机、无返回等情况，可认为该程序已成功满足所需要求。

**3.3 多组测试**

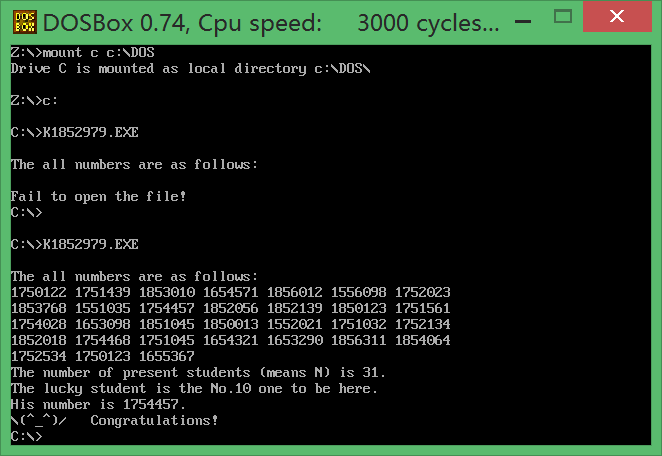
考虑到本设计题目并不需要用户从键盘输入任何信息，所以并无临界数据、等价数据等相关的测试要求。但考虑到本设计题目包含“随机数”这一要素，故应在测试时将该程序运行多次，观察是否满足题目要求“随机抽中中奖学生”的要求。

多次运行后，选取一些具有代表性的运行结果截屏如下：

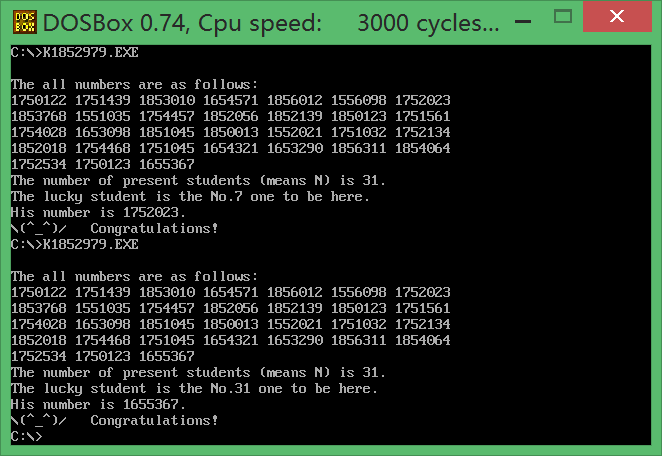
运行结果一（即上述连接时的运行结果）：



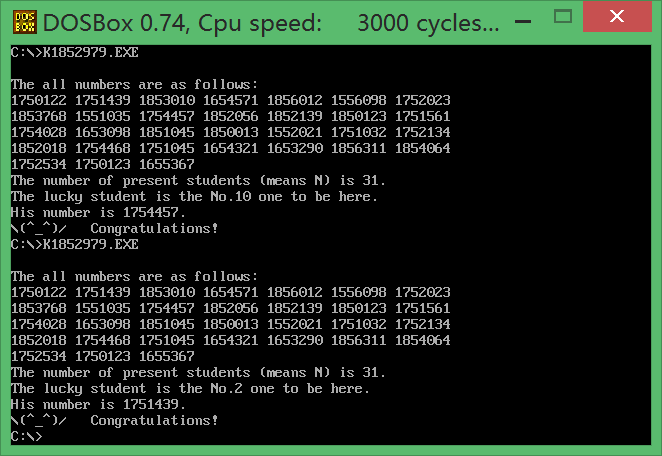
运行结果二：



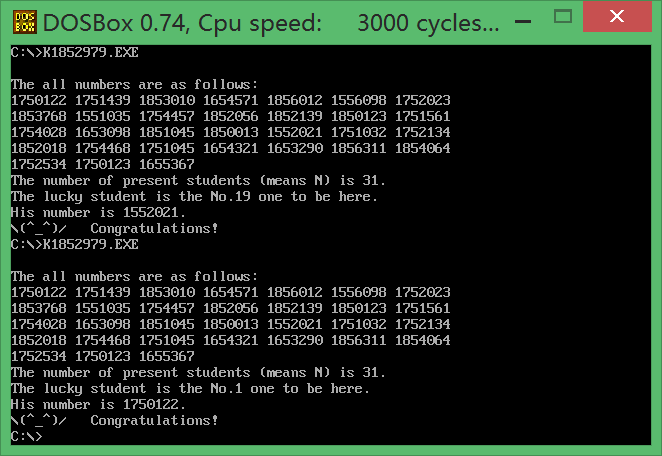
运行结果三：



运行结果四：



运行结果五：



**3.4 结果分析**

由上述运行结果，发现中奖学生的序号满足题目要求“随机抽取”，而且每组学生的序号与其对应的学号均能匹配无误，可以认为该程序的设计能够满足题目要求。

同时，经过对该程序的多次运行，发现序号排最前的学生（见运行结果五）和序号排最后的学生（见运行结果三）均有可能成为中奖学生。故可认为该程序的随机数生成算法的确能有效生成介于[1,N]的随机数，即使是边界的两个数也能有机会被取到。

此外，观察中奖学生的序号，发现不论是一位数（见运行结果四、五）还是两位数（见运行结果一、二、三）均能以十进制的形式正确输出。且一位数也能做到只输出各位，并不出现前导零的情况。故可认为该程序能够正确进行数值的十进制输出。

**四、使用说明：**

**4.1 环境说明**

本程序是8086汇编语言环境下，在DOSBox0.74可以编译运行。

**4.2 适用范围**

本程序默认可适用于所有100人以内，学号为7位阿拉伯数字的高校、学院进行人数统计，并进行抽奖。默认人数与学号位数均可通过对本程序源代码进行小规模改动，即可适应更为广泛的用户需求。具体改动方式详见使用说明4.5节。

**4.3 使用方法**

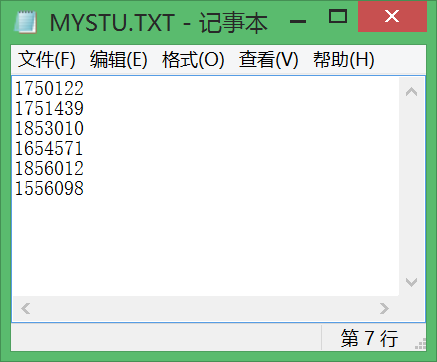
在8086环境下，将源代码.ASM文件通过DOSBox0.74汇编连接，具体连接要求见3.2节连接情况的具体说明。同时应将储存有对应学号信息的MYSTU.TXT存放于C:\目录下（如使用DOSBox0.74，则应将.TXT文档存放于DOS文件夹下；如使用其他编译工具，则应参照对应编译工具的要求存放.TXT文档）。

完成上述操作后，即可通过运行连接后的.EXE程序使用本程序。

**4.4 格式要求**

本程序默认学生总数不超过100人，学号为7位阿拉伯数字。本程序要求使用记事本等方式在C:\目录下存放一MYSTU.TXT文档，该文档内每行输入一个学生的学号，输入完成后键入“Enter”回车在下一行继续输入下一位学生学号。MYSTU.TXT文档中无需特殊的结束读取标志，键入学号完毕后保存并关闭.TXT文档即可。

具体的MYSTU.TXT文档格式可以参照如下截图：



**4.5 调试说明**

为了使本程序能适应更为广泛的用户需求，特作出如下默认程序调试说明：假使贵校学生学号为X（默认值为7）位，与会人数可能最大值为Y（默认值为100），且希望每行显示的学号总数为W（默认值为7）。则应对源程序作出如下改动：

1）将源程序第3行“900 DUP (?)”中的900改为十进制数Z1，其中Z1的计算式如下：

；

2）将源程序第75行“CMP I,7”中的7改为十进制数X；

3）将源程序第116行“CMP J,7”中的7改为十进制数W；

4）将源程序第157行“MOV CX,900”中的900改为十进制数Z1；

5）将源程序第236行“MOV DH,9”中的9改为十进制数Z2，其中Z2的计算式如下：

 ;

6）将源程序第239行“MOV CX,7”中的7改为十进制数X；

特别提醒：

1）由于本程序的PRINN子程序仅设计了[0,999]区间内十进制数的输出，所以应保证Y的值小于1000，即与会人数最高不应超过999人。如若有更高位次的人数需求，则应改写PRINN子程序，使之能适应四位乃至以上位数的十进制输出需求。

2）考虑到随着人数的增加，堆栈SSEG的大小问题，当Z1的值过大时，应当考虑增加堆栈SSEG的大小（默认为512字节），使得程序能够正常运行。

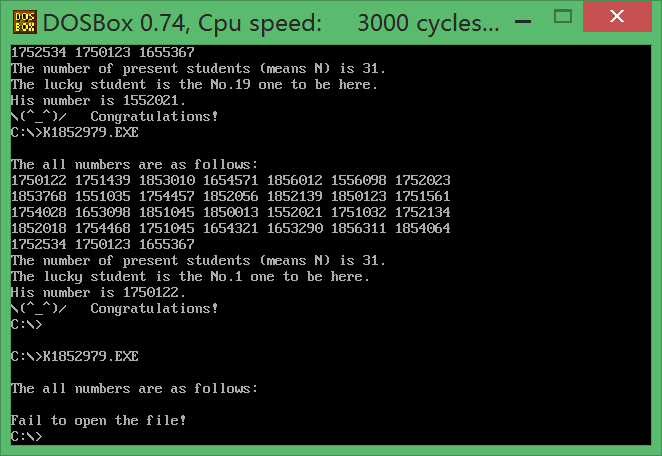
具体操作方式为更改源程序第20行“200H DUP(0)”中的200H，可以将200H的值根据所需大小进行适当调节。

**4.6 报错说明**

考虑到本程序并不需要用户从键盘输入数据，所以报错提醒均集中在文件操作方面，以下将对三种报错情况进行说明：

1）Fail to open the file!（未能成功打开文件）

报错示意截图如下：

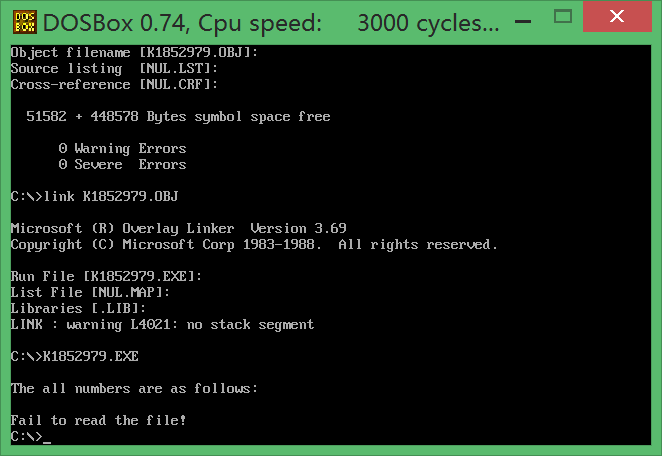


如出现上述情况，则应检查是否将MYSTU.TXT文档存放于C:\目录下，文档名是否因中英字符不同、额外添加空格等因素受到改动。

此外还有可能是管理员权限未授予.EXE，或是C:\目录所在位置权限等级较高，致使连接后的.EXE文件无权访问MYSTU.TXT文档，此时应考虑人工调节权限等级或是改换目录。

2）Fail to read the file!（未能成功读取文件）

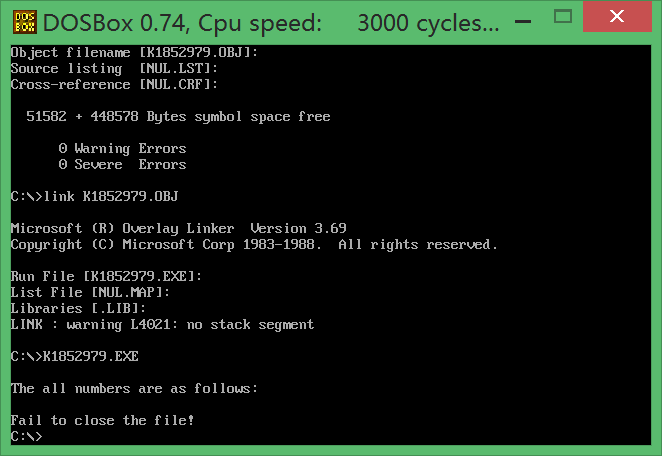
报错示意截图如下：



如出现上述情况，则应检查是否由于MYSTU.TXT文档内储存的内容过多，致使缓冲区无法完全容纳文档的全部内容，致使未能成功读取文件。

3）Fail to close the file!（未能成功关闭文件）

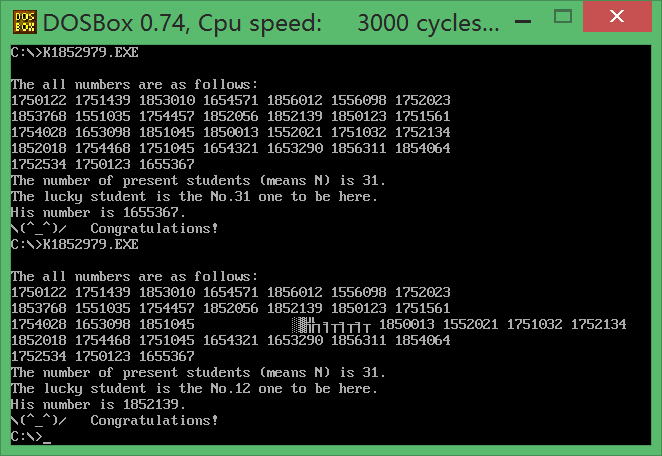
报错示意截图如下：

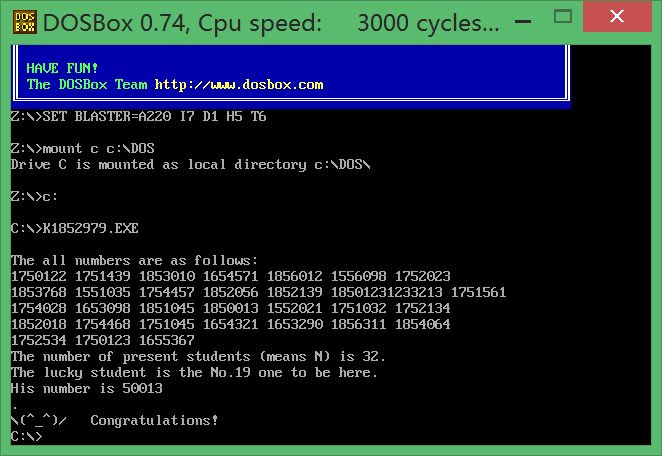


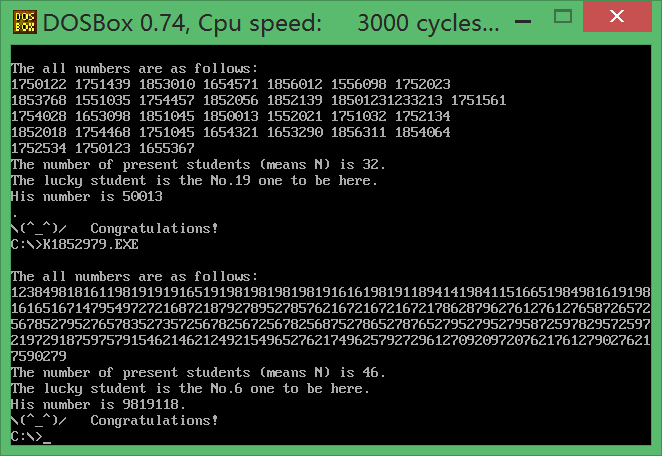
文件关闭操作大部分情况下不会报错，网上目前也少有文件关闭出错的情况分析。如出现上述情况，则建议检查DOS的文件命令是否被改写，致使程序未能正确执行文件关闭操作。

**4.7 注意事项**

本程序并未设计MYSTU.TXT文档中学号输入格式是否正确的检查环节，故若.TXT文档未能按4.4节的格式要求输入，程序并不会报错，但在显示时可能会出现一些无法预估的情况，例如出现乱码、中奖学生序号与学号不对应、学号输出错误等情况。以下几张截图均显示了.TXT文档未能正确输入所导致的错误输出情况。









因此，为了能使本程序正确输出所需结果，请用户在运行本程序前务必检查MYSTU.TXT文档的输入情况，确保与4.4节的格式要求情况相同。

**五、总结：**

**5.1情况说明**

**5.1.1 基本情况说明**

本程序完全由本人独立编译，未与其他同学合作编译，考虑到大作业要求没有提及，故未对源代码进行上报查重，但考虑到本程序源代码除自学内容外，其余功能的实现均是根据上课所讲内容自行编译得来，故代码重复率应该不会很高。

**5.1.2 自学内容说明**

自学内容主要在于汇编语言中文件的读写操作与汇编语言中通过1AH时钟获取随机数两个部分，这两部分内容均自学自网络，并无参考书目。自学内容网络链接粘贴如下：

1）汇编语言中文件的读写操作：

CSDN论坛：https://bbs.csdn.net/topics/391939160

百度知道：https://zhidao.baidu.com/question/42791488.html

2）通过1AH时钟获取随机数：

CSDN博客：https://blog.csdn.net/dingxiaowei2013/article/details/8253164

百度知道：https://zhidao.baidu.com/question/710655029287246405.html

**5.1.3 上机时间说明**

**5.1.4 上机地点说明**

**5.1.5 独创内容说明**

本次大作业程序的编写自认为中规中矩，并未包含能令人眼前一亮的思路。本程序最大的独创点主要在于：

1）网络学习资料给出的程序示例为逐字节读取，但为了将文件操作改换为更熟悉的数据段指针操作，本人独创使用了“先全部读入BUFFER，再进行操作”的方式来编写源程序。

2）题目并未要求输出全部学生的学号，但为了使中奖学生序号与学号是否对应检查起来更为方便，本人独创使用了“在公布中奖学生前输出所有学生学号”的方式。

3）题目并未要求给出不同学号长度、不同规模预计人数情况下程序如何改写的说明，但为了使本程序的说明能更具有普适性，本人独创使用了“给出小规模改动的方式说明，从而增大适用范围”的使用说明编写方式来编写4.5节，希望能起到更为普适的作用。

**5.2 心得体会**

**5.3 问题分析**

**5.3.1 程序改进分析**

受限于本人有限的汇编知识储备，本次大作业设计的程序还包含着诸多不足，我认为程序主要存在的可改进的地方有以下几点：

1）该程序并不包含监测MYSTU.TXT文档是否符合4.4节所规定格式要求的功能，可能会导致诸如4.7节所展示的输出乱码依旧不报错的情况。解决方式可以考虑在主程序调用OPENN函数后，调用COUT函数前，调用一个包含监测功能的子程序，用于核查MYSTU.TXT文档是否对学号录入正确。

2）考虑到用户可能会希望调节学生学号的长度、与会学生的最高限额等，可以考虑制作一个面向用户的前端界面，可供用户输入学生的学号长度等变量。同时还可以进一步考虑并入“输入学号”的功能，让用户能够在同一个前端界面上完成与会学生学号的录入、抽奖、显示等功能。

**5.3.2 本人问题分析**

**六、签字：**