**2022-2023学年第二学期《汇编语言程序设计实验》期末大作业报告**

姓名：吕万朗 学号：2021103288 专业：网络空间安全

1. 任务要求

利用8086汇编语言编写简要成绩管理系统。以本课程为例，具体要求如下。

1. 该系统能提供信息录入、查询、最终成绩排序功能及分数统计信息显示（各个总成绩分数段的人数、平均分、最高分、最低分）。

2. 信息录入包括：学生的姓名、学号、及16次作业成绩和一次大作业成绩。（通过键盘输入）

3. 该系统能自动计算学生的最终成绩，按照平时作业成绩占40%，大作业成绩占60%计算。

4. 查询功能的查询方式需在显示界面有提示是按姓名查询还是学号查询。

5. 成绩排序默认按照从低到高排序。排序显示要求同时显示学生的姓名、学号、平时成绩、大作业成绩、总成绩、及排名。

6. 各个总成绩分数段的人数的统计按照：90-100一个分数段；80-89一个分数段；60-79一个分数段；60以下一个分数段。

1. 代码思路及程序框架

整个程序代码分为三个部分，data段、宏定义部分、code段。其中data段中放置我们的菜单以及一系列需要输出的提示信息、提醒标号等等；宏定义主要定义了一些比较常用的字符串输入、输出，成绩输入错误检查、排序等一些需要大量自动运行的模块；code段放置了6大模块，是代码的运行核心。

一、data段定义一个名为LIST的字节数组，用于存储程序的菜单和提示信息，其中data\_s最为重要，存储了主要学生主要信息。还定义名为hint1、hint2、hint\_lastgrade、hint3、hint4、hint5、hint6等变量，用于存储各种提示信息。类似缓冲区，方便后续平均成绩等的计算，还想着在data段定义student\_final\_score、student\_level、lev1、lev2、lev3、lev4、high\_score、low\_score、等变量，用于存储学生最终成绩、分数段信息、分数统计信息等。

二、在宏定义的区域，定义一个名为shuru的宏，用于从键盘读取单个字符并回显；定义一个名为shuchu的宏，用于输出字符串；定义一个名为lastgrade\_numcheck的宏，用于检查大作业成绩的输入是否正确；定义一个名为gr\_numcheck的宏，用于检查平时成绩的输入是否正确；定义一个名为exchange\_stu的宏，用于对学生成绩进行排序时进行交换。这些宏定义的主要作用是简化成绩管理汇编程序中常用的操作，减少编写程序的难度和代码量。

三、code段：

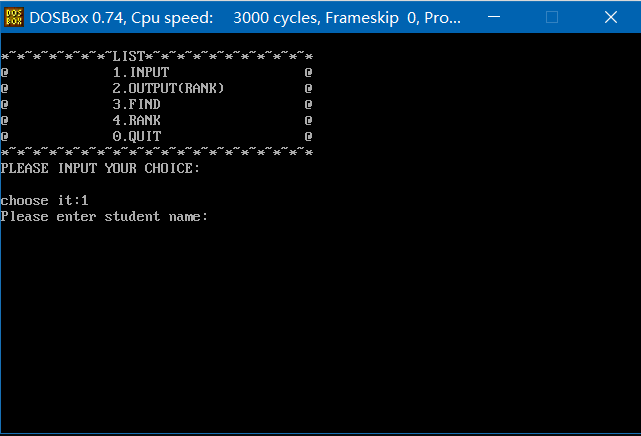
1. choice子程序：调用宏输入子程序，检测输入的数字，并跳转到对应的子功能程序去执行，输出LIST菜单。其中在跳转之前，使用call命令还能在call之前保存一些必要的值，方便循环操作等。
2. in\_put子程序：就是input程序，用于存储输入的学生信息，bx寄存器保存了标号data\_s的地址，每个学生的信息录入完毕后，循环结束，每次循环都跳到data\_s段的下一行，bx+105；同时检查空格，是否要输入下一个成绩。
3. ou\_put子程序：就是output程序，调用宏定义输出LIST和提示信息。
4. ra\_nk子程序：就是rank程序，排序的同时，加上分数段统计和平均分输出（首先要使用cal子程序计算好学生成绩）。
5. ra\_nk过程：①将数据段地址赋给DS寄存器，输出成绩排序提示信息；②使用循环来比较学生成绩并进行排序，这里使用的是冒泡排序法，循环次数为n-1，其中n为数据总行数；③对于每个学生的成绩，将其百位、十位、个位依次存入缓冲区，并计算出最终成绩；④比较前一个学生的总成绩与下一个学生的总成绩，如果前一个学生的总成绩小于下一个学生的总成绩，则交换两个学生的数据，循环结束后调用rank\_output过程进行输出。
6. rank\_output过程：①将数据段地址赋给DS寄存器，并使用循环输出所有学生的姓名和成绩。②对于每个学生，将其平时成绩、大作业成绩和最终成绩依次存入缓冲区。③使用shuchu宏定义输出缓冲区中的数据。
7. rank\_last\_output过程：则是将最后缓冲区的内容放到需要显示的字符串上，并输出该字符串。
8. **cal\_culate子程序**：也就是calculate程序，用于计算学生的平时成绩，最终期末成绩，按照平时作业成绩占40%，大作业成绩占60%计算，将存储在data\_s中的字符串转化为数字，并按照百位\*100，十位\*10，个位\*1的规律加到缓冲区存储，最终加完的数再进行\*0.4和\*0.6并相加，得到的期末成绩，再将其转换为字符串，保存回到data\_s中，其他计算平均分、排序等子程序都用到了这个思路。

其中还有两个子程序，内容较为独立，分开列出，但还是属于cal子程序。

1. grade\_levels子程序：对学生的成绩进行区域划分并且输出显示，相较于其他模块不同的是这里只用到了期末成绩进行划分，不同于排序，只是用了各层类似if和else语句来进行划分，并统计各个分层人数，最后输出显示。
2. average\_and\_hilo子程序：对全体学生平时成绩、大作业成绩、期末成绩进行平均分、最高分和最低分的统计计算和处理，并显示出来。
3. search子程序：姓名和学号同理，根据姓名或者学号的选择，来进行匹配，将输入的字符保存到缓冲区，每个字符依次和学生信息中的字符比较，若有不相同，则进行下一个学生的比较，直到选到学号或姓名完全相同位置（空格也进了比对，但是不妨碍结果）。如果全部学生都不匹配，则输出错误提示信息；若有匹配，则输出该学生全部成绩信息。
4. show子程序：非核心程序，用于主菜单的回显和让程序按0之后不会草草结束。
5. 模块功能正确性测试

考虑到测试效率问题，每个模块的测试使用的data段都大致相同，后面复杂模块的测试在不影响测试正确性的情况下，含有前面模块的部分功能或内容。

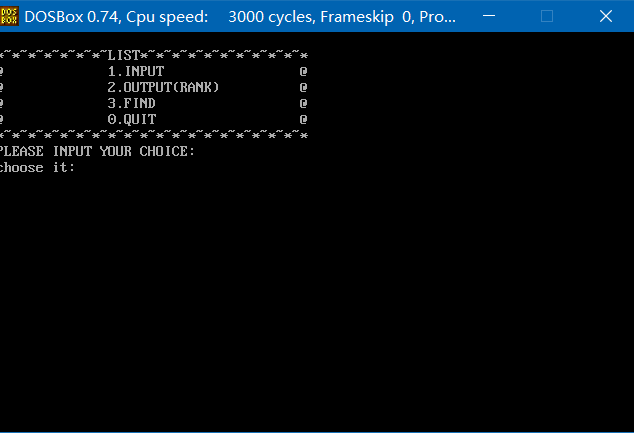
输入宏定义：



在菜单选项子程序中能够一并测试使用，并且在多个地方都使用到，正确性有保证

此处输入1，立马跳到input模块，开始要求输入学生名字

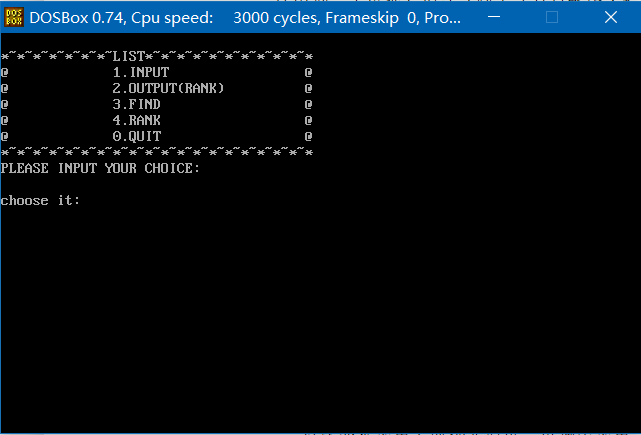
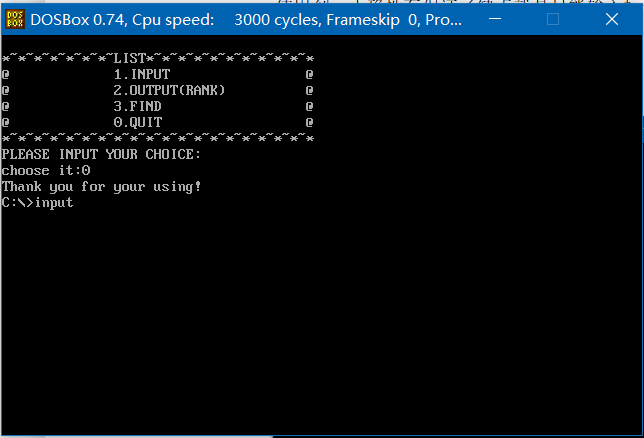
输出宏定义：



菜单的显示和其他提醒语句都使用输出宏定义进行输出

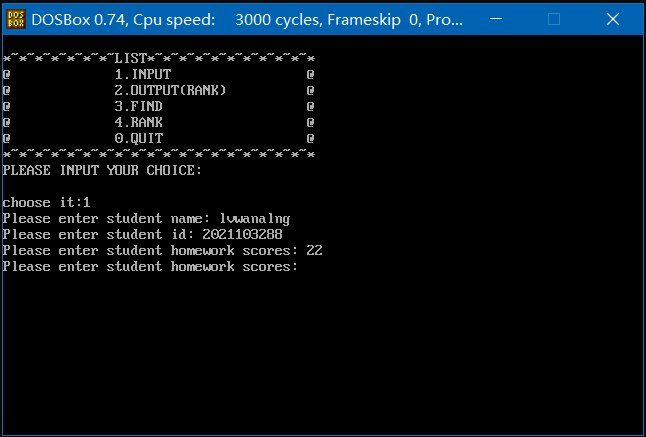
和输入宏定义类似，最开始在菜单选项子程序中能够一并测试使用，并且在多个地方都使用到，正确性有保证（缺点就是只能输入标号进行对应的输出）

菜单显示子程序：

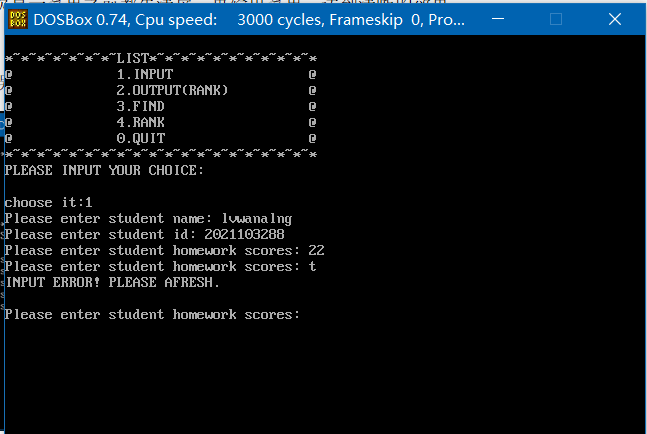


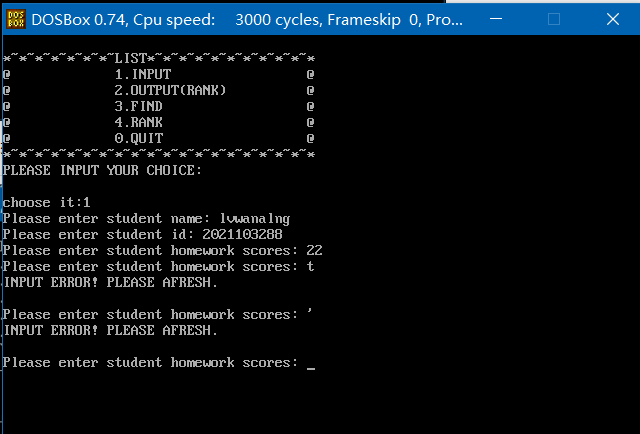
每次显示菜单之前都先清屏，再输出菜单，达到清晰的效果

平时成绩输入错误检查宏定义：



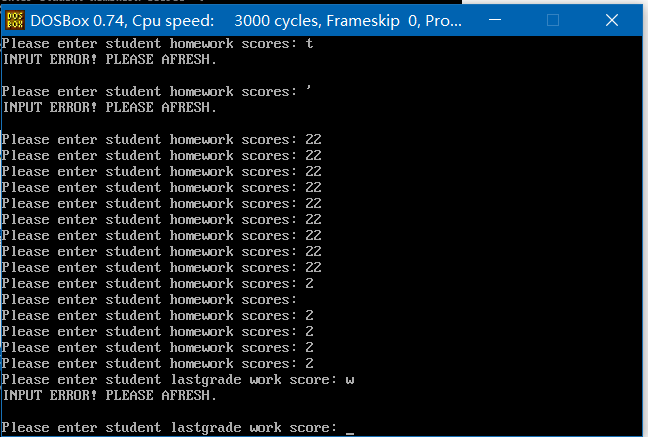
当输入正确的成绩的时候，即输入数字而非字母，不会报错



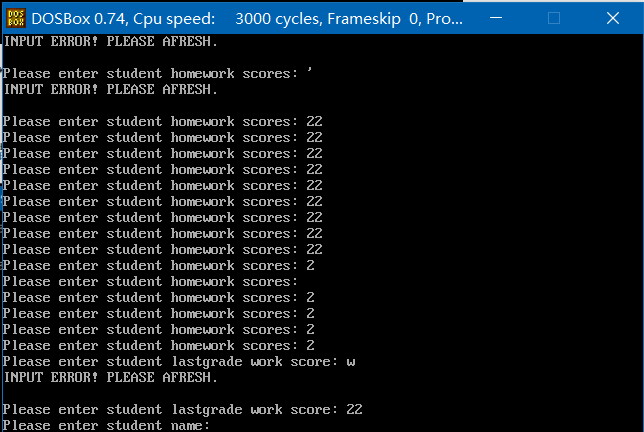


一旦输入字母或者其他符号的时候，就会报错，弹出INPUT ERROR！

大作业成绩输入错误检查宏定义：

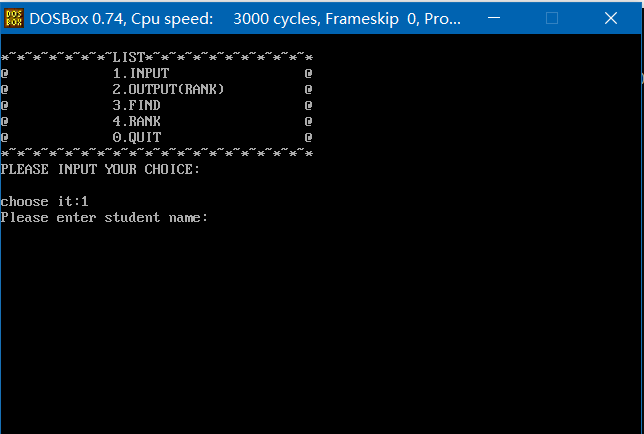


和上面的宏定义类似，输入字母或者其他符号的时候就会报错

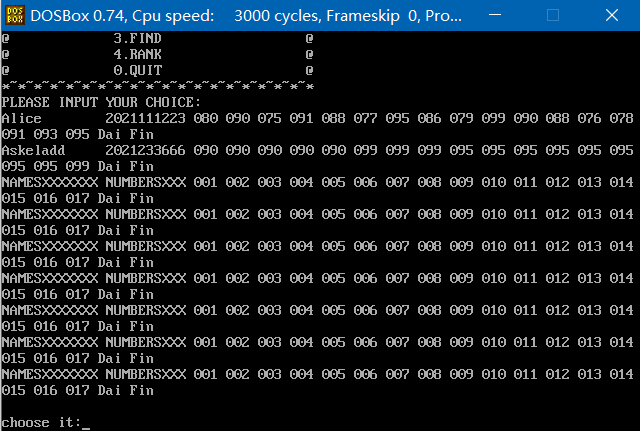


输入数字则无报错，程序正常运行，此时进入下一个学生的信息输入

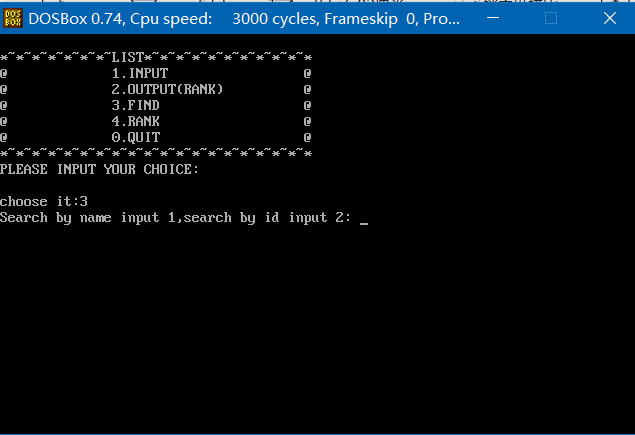
菜单选项子程序：



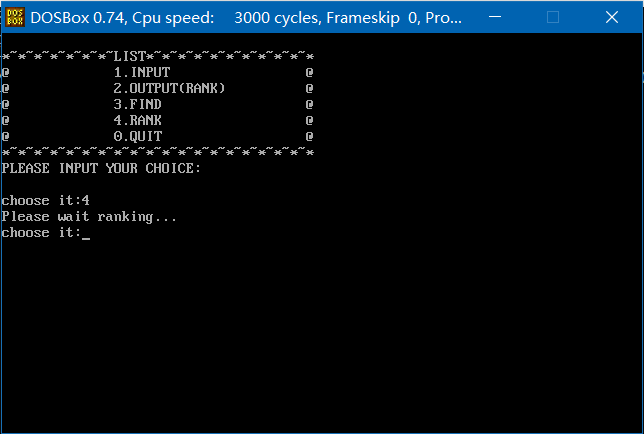
按1进入输入模式



按2输出存储信息



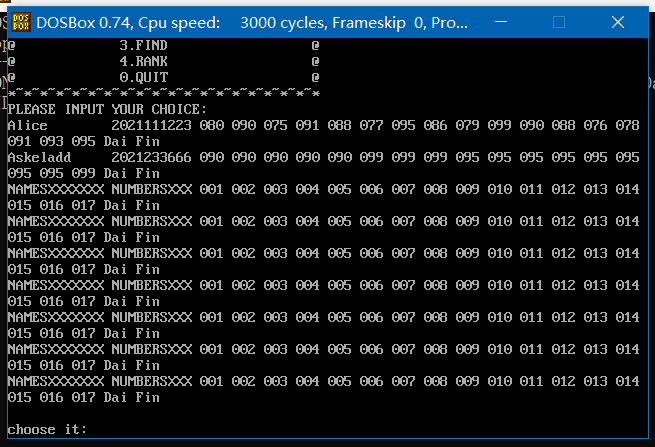
按3进入查询模式（这里只是输出一段文字，代表进入了模块）



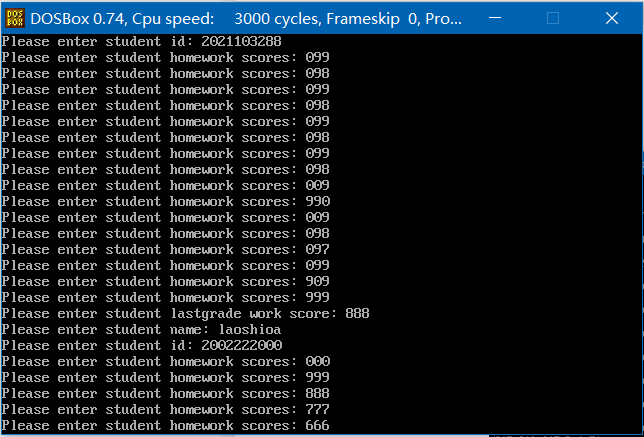
按4进行成绩排序

按0退出

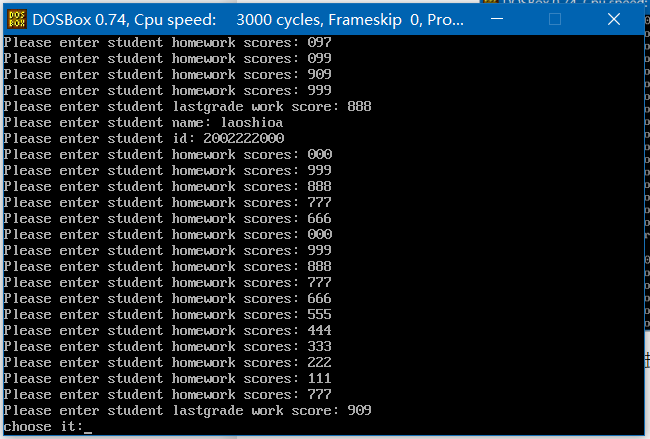
学生信息输入模块：



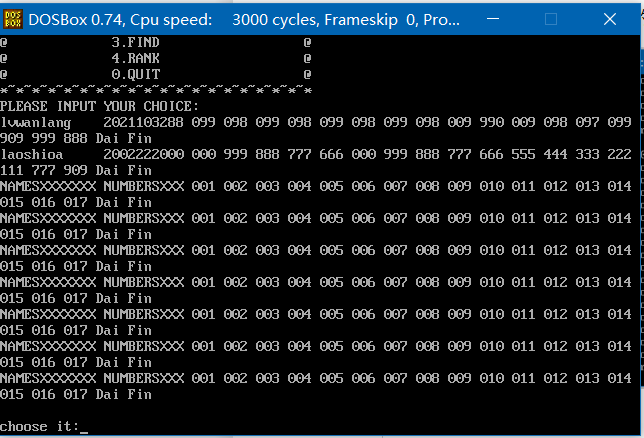
输出查看目前存储信息



陆续输入每个学生的信息（模块测试只输入了两名学生信息）

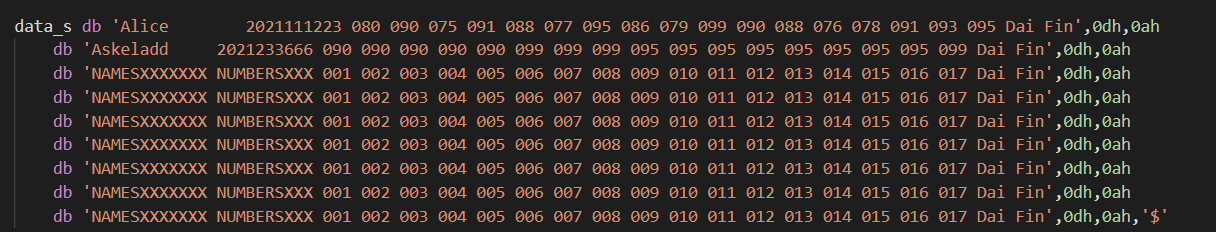


输入完毕

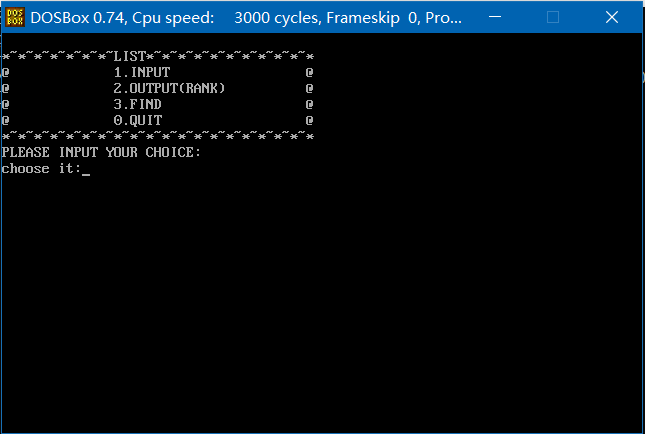


再次使用输出模块功能，输出存储学生信息，前两名学生信息被录入（改变）

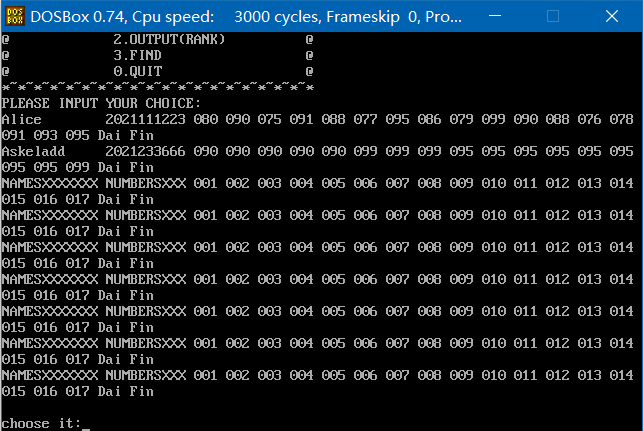
存储信息输出模块：



存储的信息如上

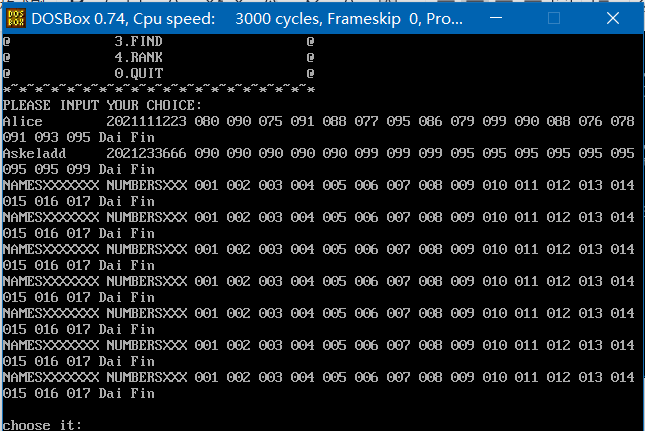


开始表单

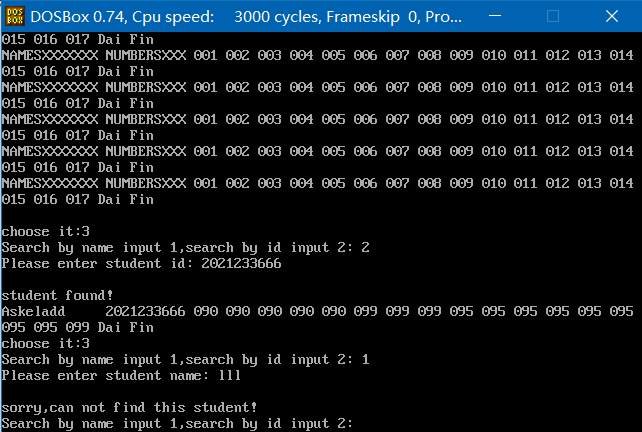
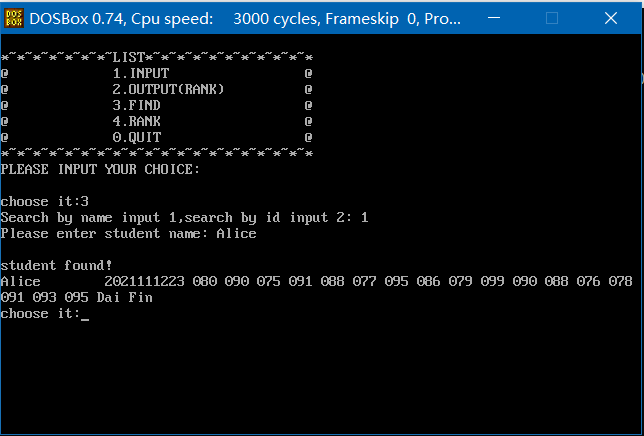


按下2，输出存储信息成功

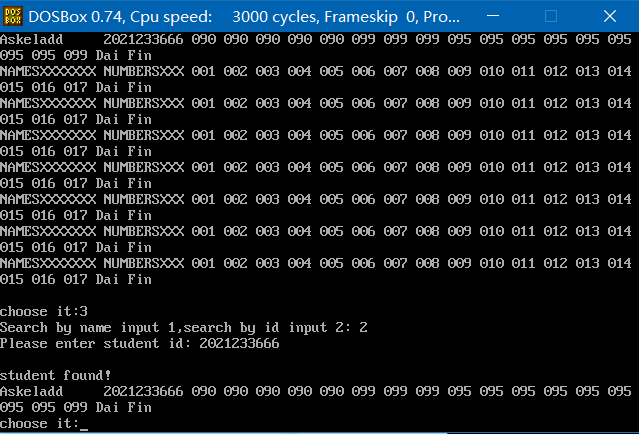
存储信息查询模块：



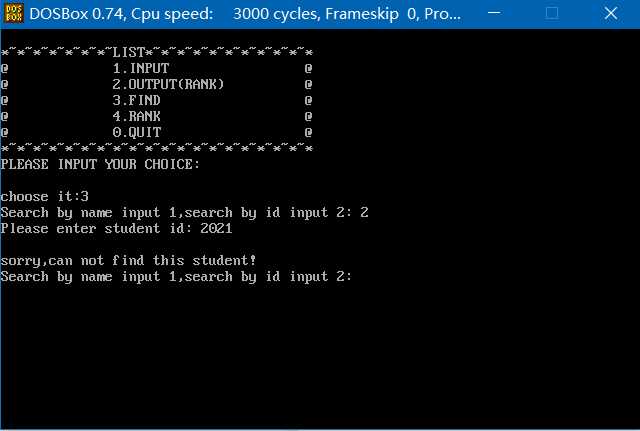
首先先查看目前存储的信息，Alice学生和Askeladd学生



按姓名查询Alice，查询成功，输入错误姓名，查询失败

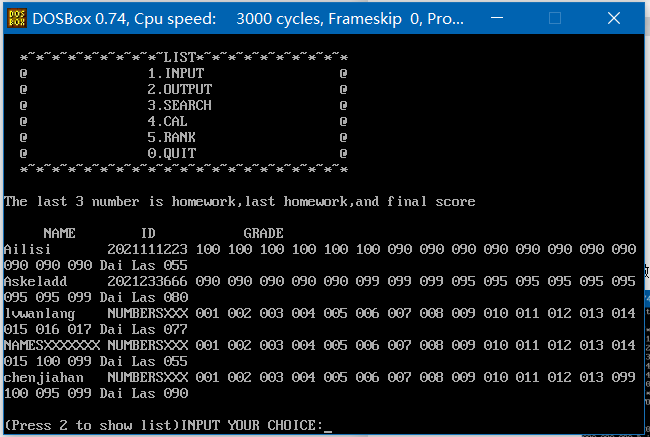


按学号查询2021233666，查询成功

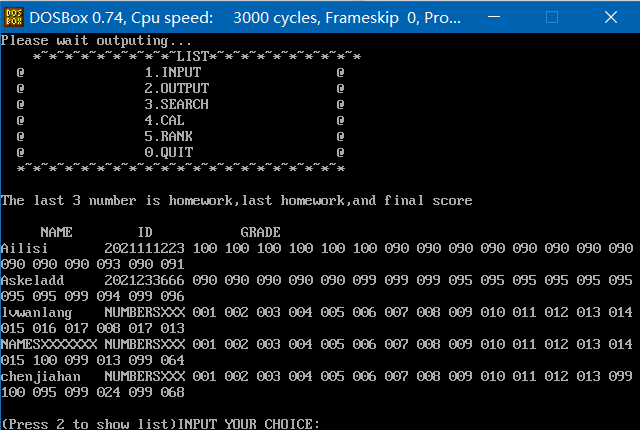


输入错误学号，查询失败

成绩计算模块：

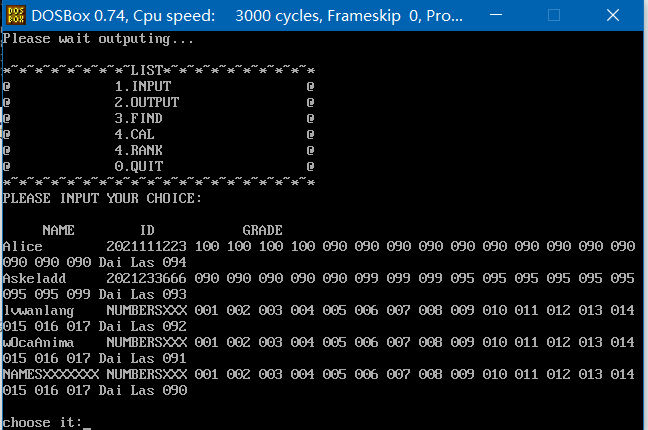


显示原本保存的成绩

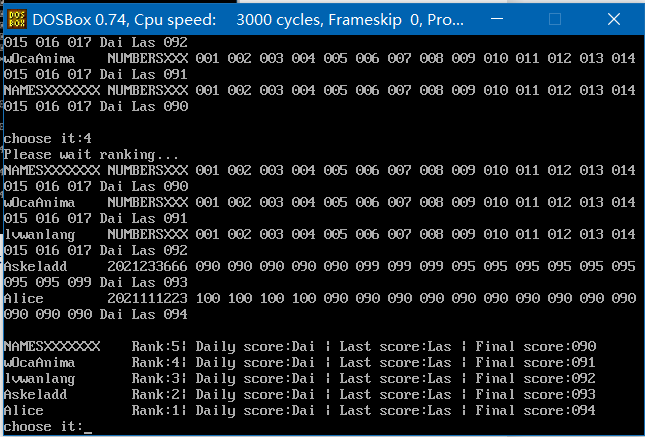


按4之后，再点击输出存储信息，查看计算好的平时成绩和最终成绩

最终成绩排序模块和分数信息统计显示模块：



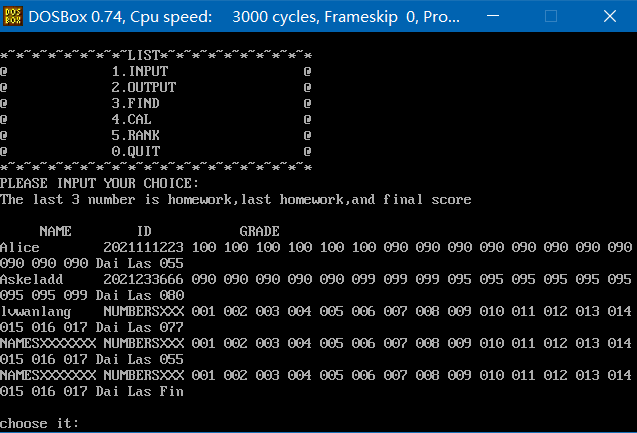
原本保存的成绩



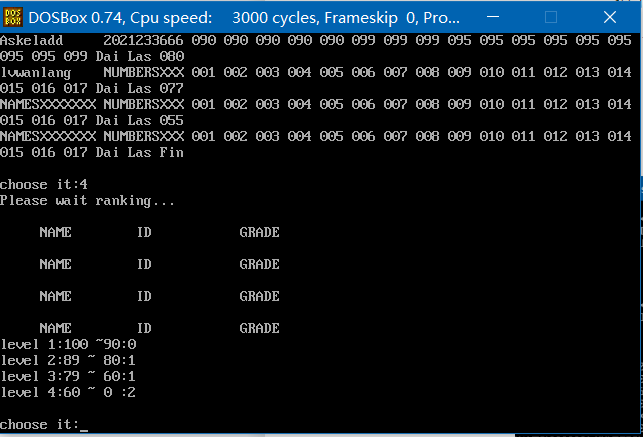
摁下4之后，按照成绩由低到高排序

并输出学生姓名、学号，平时成绩、大作业成绩、以及最终成绩和排名

成绩分段显示模块：

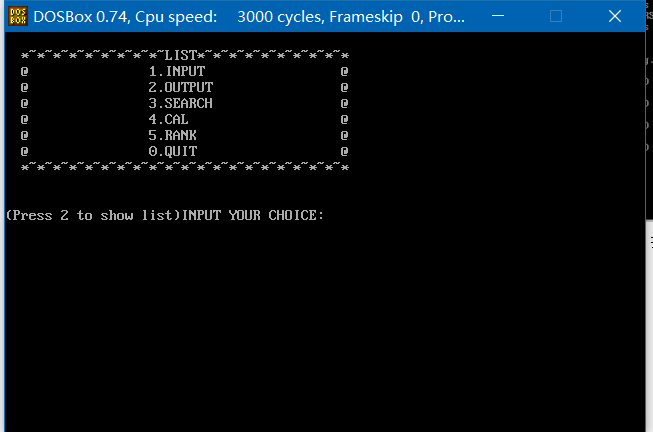


原本的成绩是，两个不及格，一个80，一个77

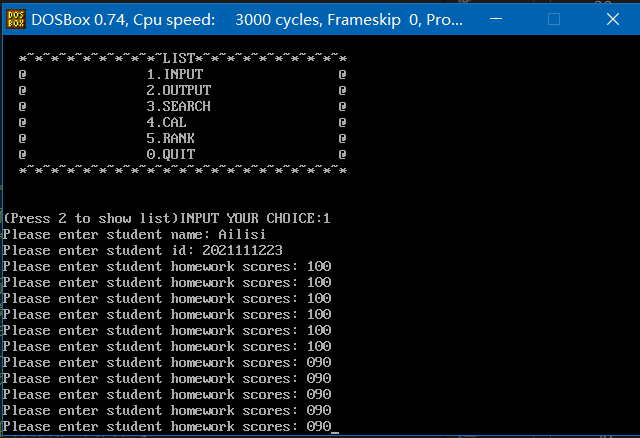


排序并统计之后，显示分段人数

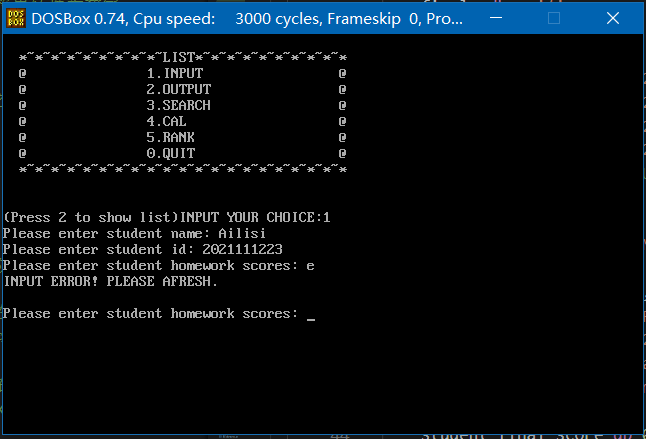
1. 集成测试



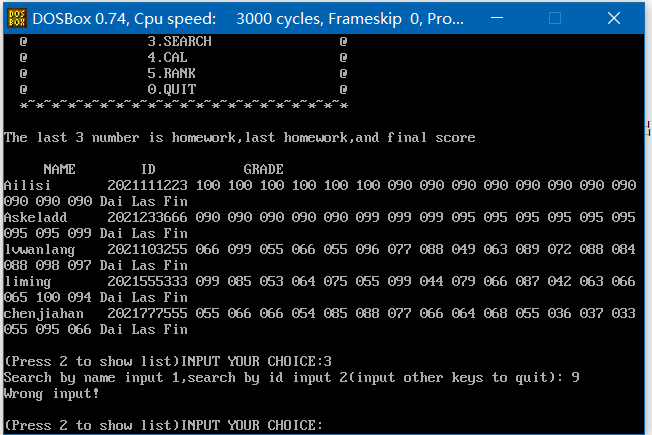
打开成绩管理系统程序，显示主页面，按1输入



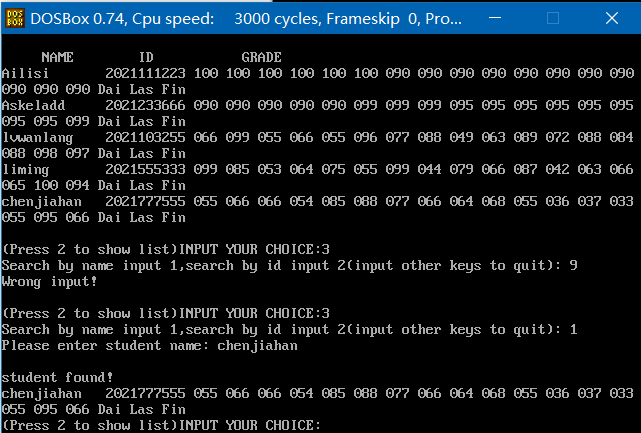
开始输入模式，输入学生信息（不全显示，其他输入省略）



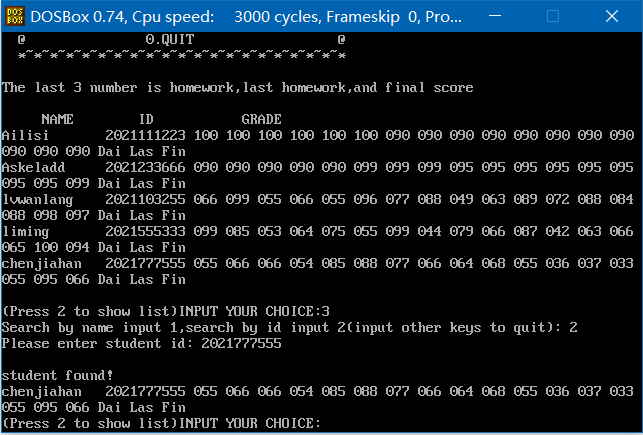
成绩输入纠错，如果检测到有字母或其他符号输入就报错并且让你重新输入



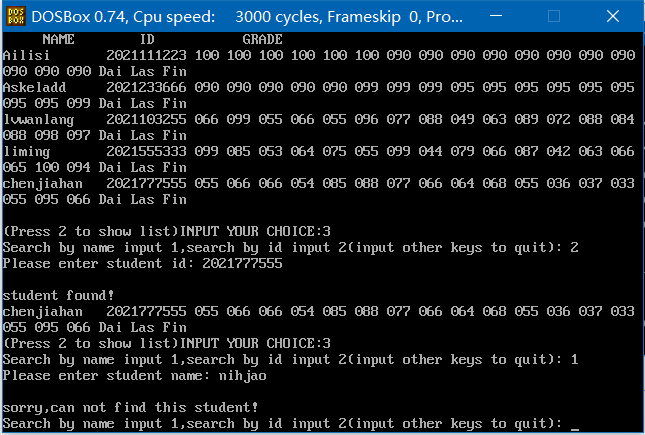
输入2查看刚刚输入的成绩，点击3进入查询模式，此处选择9，退出查询模式



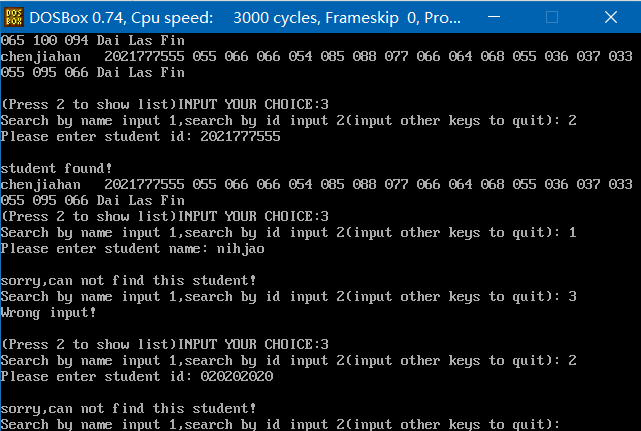
再次点击3，进入查询模式，点击1按姓名查询，输入正确姓名，查询成功



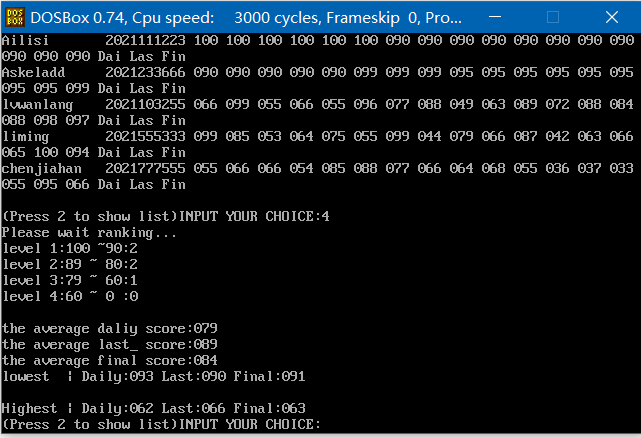
再次点击3，进入查询模式，点击2按学号查询，输入正确学号，查询成功



点击3，进入查询模式，点击1按姓名查询，输入错误姓名，查询失败



再次点击3，进入查询模式，点击2按学号查询，输入错误学号，查询失败



返回主菜单（点击2,），点击4，计算成绩，自动计算完毕

输出各个分段人数

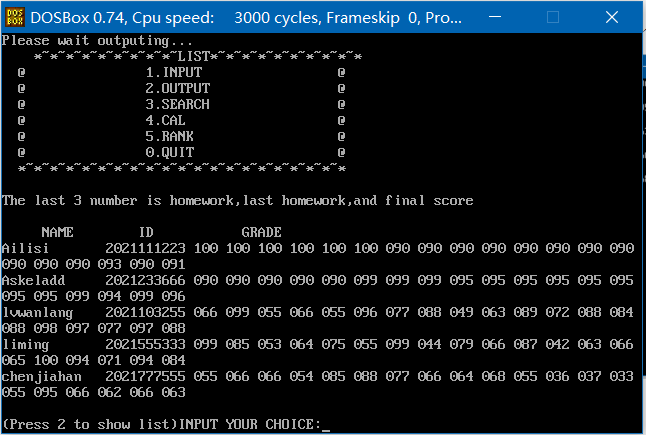
这里是90-100区间2人，80-89区间2人，79-60区间1人，60以下区间0人

（具体计算出来的分数需要再次点击2，返回主菜单并输出学生各项信息查看）

输出平时成绩、大作业成绩、总成绩平均成绩

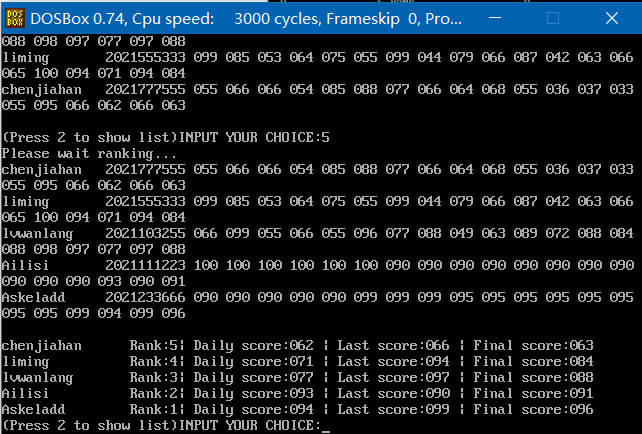
自动计算学生的最终成绩，按照平时作业成绩占40%，大作业成绩占60%计算

输出最高，最低分分数



再次点击2，返回主菜单，查看到刚才计算好的成绩平均数

每个学生后三个成绩分别就是平时平均成绩、大作业成绩和期末总成绩



点击2返回主菜单，点击5，开始排序功能

先输出了存储的学生信息，并按照期末成绩从低到高给学生排序，输出排名

1. 系统的缺陷分析

1.在成绩输入的时候，如果有误操作，不能输入backspace删除错误输入，只能重新输入该成绩。

2.在主菜单输入非1、2、3、4、5的数字就会报错，误操作就会导致立马跳出程序

3.宏定义较少，所以如果有要求更多学生保存信息，更改代码的复杂度高

4.计算稳定性还需要增加，如果遇到更多学生的信息需要输入，计算的平均数结果可能会不稳定

5.代码基本是纯手敲，（宏定义少）可能会有一些遗漏之处没有注意到

1. 源代码

assume cs:code,ds:data

data segment

    LIST db 13,10

            db '  \*~\*~\*~\*~\*~\*~\*~\*~\*~LIST\*~\*~\*~\*~\*~\*~\*~\*~\*~\*',13,10

            db '  @               1.INPUT                 @',13,10

            db '  @               2.OUTPUT                @',13,10

            db '  @               3.SEARCH                @',13,10

            db '  @               4.CAL                   @',13,10

            db '  @               5.RANK                  @',13,10

            db '  @               0.QUIT                  @',13,10

            db '  \*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*~\*',13,10

            db 13,10,'$'

    student db '            ','$'  ; 学生姓名，最多12个字符，以0结束

    id db '          ','$'     ; 学生学号，最多10个字符，以0结束

    hos db '090 090 090 090 090 090 090 090 091 092 093 091 090 090 094 092 $'

    hos\_total db '099 ','$'

    lastgrade dw 0,'$'         ; 大作业成绩

    total db 0,0        ; 总成绩

    huiche db 13,10,'$';一个回车

    ave\_total dw 0,'$'

    rankis db 'rank: ',0,'$'

    final1 db 0,'$'

    final2 db 0,'$'

    data\_s db '                                                                                            Dai Las Fin',0dh,0ah

        db '                                                                                            Dai Las Fin',0dh,0ah

        db '                                                                                            Dai Las Fin',0dh,0ah

        db '                                                                                            Dai Las Fin',0dh,0ah

        db '                                                                                            Dai Las Fin',0dh,0ah

        ;db 'NAMESXXXXXXX NUMBERSXXX 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 Dai las Fin',0dh,0ah

        db '$'

    hint1 db 13,10,'     NAME        ID           GRADE',13,10,'$'

    hint2 db 13,10,'student found! $'

    hint\_lastgrade db 13,10,'lastgrade: ',13,10,'$'

    hint3 db 13,10,'INPUT ERROR! PLEASE AFRESH.',13,10,'$'

    hint4 db 13,10,'(Press 2 to show list)INPUT YOUR CHOICE:','$'

    hint5 db 13,10,'sorry,can not find this student!','$'

    hint6 db 'The last 3 number is homework,last homework,and final score',13,10,'$'

    student\_final\_score db 0,0,0,0,0      ;存储学生最终成绩

    student\_level db 0,0,0,0,0,'$'            ;存储所在分段学生个数

    lev1 db 'level 1:100 ~90:  ',13,10,'$'

    lev2 db 'level 2:89 ~ 80:  ',13,10,'$'

    lev3 db 'level 3:79 ~ 60:  ',13,10,'$'

    lev4 db 'level 4:60 ~ 0 :  ',13,10,'$'

    high\_score db 'Highest | Daily:    Last:    Final:    ','$'

    low\_score  db 'lowest  | Daily:    Last:    Final:    ','$'

    av\_da db 'the average daliy score:   ',13,10,'$'

    av\_la db 'the average last\_ score:   ',13,10,'$'

    av\_fi db 'the average final score:   ',13,10,'$'

    average\_daily dw 0,'$'

    average\_last dw 0,'$'

    average\_final dw 0,'$'

    ; 提示信息

    name\_prompt db 13,10,'Please enter student name: $'

    id\_prompt db 13,10,'Please enter student id: $'

    home\_prompt db 13,10,'Please enter student homework scores: $'

    lastgrade\_prompt db 13,10,'Please enter student lastgrade work score: $'

    total\_prompt db 13,10,'Final score: $'

    search\_prompt db 13,10,'Search by name input 1,search by id input 2(input other keys to quit): $'

    rank\_prompt db 13,10,'Please wait ranking...',13,10,'$'

    output\_prompt db 13,10,'Please wait outputing... ',13,10,'$'

    thanks db 13,10,'Thank you for your using! $'

    WARNING db 13,10,'Wrong input! $'

    after\_cal db 13,10,'                Rank: | Daily score:    | Last score:    | Final score:    ','$'

data ends

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;功能宏定义

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;单字符输入宏定义

shuru macro

    mov ah,1    ;键盘输入并回显

    int 21h

endm

;字符串输出宏定义

shuchu macro ad

    lea dx,ad

    mov ah,09h

    ;mov dx,offset ad

    int 21h

endm

;大作业成绩输入错误检查宏定义

lastgrade\_numcheck macro alf

    cmp alf,20h  ;检查空格

    je las\_con

    cmp alf,'9'

    ja ierror

    cmp alf,'0'

    jb ierror

    jmp las\_con         ;没错就继续

ierror:

    shuchu hint3        ;提示输入错误;跳回重新输入

    jmp  lastgrade\_shuru

endm

;平时成绩输入错误检查宏定义

gr\_numcheck macro ale

    cmp ale,20h  ;这里要先检查空格

    je gr\_con

    cmp ale,'9'

    ja iferror

    cmp ale,'0'

    jb iferror

    jmp gr\_con

iferror:

    shuchu hint3        ;提示输入错误;跳回重新输入

    jmp  gr\_s0

endm

;学生成绩排序交换宏定义

exchange\_stu macro

    mov dx,105

    mov si,0

chs:

    mov al,ds:[bx+si]

    mov ah,ds:[bx+si+105]     ;下一行

    mov ds:[bx+si],ah

    mov ds:[bx+si+105],al

    inc si

    dec dx

    cmp dx,0

    jne chs

endm

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

code segment

start:

    call show

    call choice

Exit:

    mov ax,data

    mov ds,ax

    shuchu thanks

    mov ax,4c00h

    int 21h

;菜单选项子程序

choice proc near;;

    mov ax,data

    mov ds,ax

    shuchu hint4

    shuru       ;调用宏输入一个字符

    cmp al,49   ;if al==1

    je  inp     ;调用输入功能

    cmp al,50   ;if al==2

    je out\_p

    cmp al,51   ;if al==3

    je sear

    cmp al,52   ;if al==4

    je calculate

    cmp al,53   ;if al==5

    je ran

    cmp al,48   ;if al==0

    je Exit

    shuchu WARNING

    ret         ;返回调用的地方

inp:

    mov cx,5            ;存储的数组有多少行，跟着in\_put的第二行的ax改动

inp\_s:

    call in\_put

    loop inp\_s

    jmp choice

out\_p:

    call out\_put

    jmp choice

calculate:

    mov ax,data

    mov ds,ax

    shuchu rank\_prompt

    mov cx,5

calculate\_s:

    push cx

    mov ax,5

    sub ax,cx

    mov bl,105

    mul bl

    mov bx,offset data\_s

    add bx,ax

    call cal\_culate

    pop cx

    loop calculate\_s

    call grade\_levels

    call average\_and\_hilo

    jmp choice

ran:

    call ra\_nk

    jmp choice

sear:

    call search

    shuchu huiche

    jmp choice

choice endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;已知一个学生所有数据的一行数组有105个字节

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

in\_put proc near

    push cx         ;关乎第几行的记录，返回前先存着

    mov ax,5                ;算出现在是第几行的操作

    sub ax,cx               ;用总行数减去当前循环的次数，等于现在第几行，保存到ax中

    mov bx,105              ;一行数组有101个字节

    mul bx                  ;乘法的结果保存在ax中

    mov bx,offset data\_s  ;先取得存储空间的偏移地址

    add bx,ax

;姓名输入

    mov ax,data

    mov ds,ax

    shuchu name\_prompt

    mov si,0                ;开头第一个字节

name\_in:

    shuru                   ;输入名字

    mov ds:[bx+si],al          ;复制字符串

    inc si

    cmp al,20h  ;检查空格

    jne name\_in

;学号输入

    shuchu id\_prompt

    mov si,13               ;定位ID的存储偏移地址,为每一行的第13个位

id\_in:

    shuru                   ;输入ID

    mov ds:[bx+si],al          ;复制字符串

    inc si

    cmp al,20h  ;检查空格

    jne id\_in

;平时成绩输入子程序

GR\_shuru:

    mov cx,16       ;16个平时成绩

    mov si,24        ;平时成绩的存储偏移地址为每一行的第24个字节位,这个要一开始就设置好了，不然后面每一次循环都要重置

gr\_s0:

    mov ax,data

    mov ds,ax

    shuchu home\_prompt

gr\_s:

    shuru           ;键盘接收一个字符

gr\_panduan:

    gr\_numcheck al      ;检查有没有输错

gr\_con:

    mov [bx+si],al      ;复制字符串

    inc si

    cmp al,20h      ;检查空格，是否要输入下一个成绩

    jne gr\_s

    loop gr\_s0

;大作业成绩输入子程序

lastgrade\_shuru:

    shuchu lastgrade\_prompt

    mov si,88               ;定位每一行第88位为大作业成绩偏移位置

las\_s:

    shuru                   ;输入大作业成绩

    lastgrade\_numcheck al

las\_con:

    mov [bx+si],al          ;复制字符串

    inc si

    cmp al,20h              ;检查空格

    jne las\_s               ;不是空格就继续接收输入

    pop cx

    ret

in\_put endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;总输出子程序

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

out\_put proc near;;

    mov ax,data

    mov ds,ax

    shuchu output\_prompt

    mov si,0

    mov bx,offset data\_s

    shuchu LIST

    shuchu hint6

    shuchu hint1

    shuchu data\_s       ;输出内存存储的学生数据

    ret

out\_put endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;成绩排序子程序

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ra\_nk proc near

    mov ax,data

    mov ds,ax

    shuchu rank\_prompt

    mov ax,data

    mov ds,ax

    mov di,offset final1

    mov bp,offset final2

    mov cx,4            ;循环次数=n-1，n就是数据总行数

cal\_final\_inputs:

    ;这里采用最简单的冒泡排序

    mov bx,offset data\_s    ;每次外循环需要重置bx，指向第一个存储位置

    push cx                 ;入栈外循环的cx

cal\_final:

    ;将字符数字存入缓冲区并比较

    mov si,0

    mov ax,0

    mov al,ds:[bx+si+100]   ;百位,100为最终成绩存储百位数的偏移地址

    sub al,30h          ;将字符串转换为数字

    mov dl,100      ;百位的数乘100

    mul dl

    add ds:[di],al      ;存储

    mov al,ds:[bx+si+205]    ;下一个学生的最终成绩的百位偏移地址，205=100+105

    sub al,30h          ;将字符串转换为数字

    mov dl,100      ;百位的数乘100

    mul dl

    add ds:[bp],al

    inc si

    mov al,ds:[bx+si+100]   ;十位

    sub al,30h          ;将字符串转换为数字

    mov dl,10       ;十位的数乘10

    mul dl

    add ds:[di],al      ;存储

    mov al,ds:[bx+si+205]    ;下一个学生的最终成绩的十位偏移地址

    sub al,30h          ;将字符串转换为数字

    mov dl,10       ;十位的数乘10

    mul dl

    add ds:[bp],al

    inc si

    mov al,ds:[bx+si+100]   ;个位

    sub al,30h          ;将字符串转换为数字

    add ds:[di],al     ;存储

    mov al,ds:[bx+si+205]   ;个位

    sub al,30h          ;将字符串转换为数字

    add ds:[bp],al      ;存储

    inc si

;\*\*\*\*\*\*\*比较，判断是否要交换

    mov al,ds:[di]      ;final1,即目前循环指到的对象,前一个学生

    mov ah,ds:[bp]      ;final2

    cmp ah,al           ;对比学生的总成绩

    ja continue\_cal\_final      ;前一个小于等于后一个就不用交换,大于就交换

    exchange\_stu                ;交换

continue\_cal\_final:

    mov al,0

    mov ds:[di],al

    mov ds:[bp],al

    add bx,105          ;进入到下一行比较

    loop cal\_final        ;loop一起消耗着内循环的cx

    pop cx                      ;出栈外循环的cx

    dec cx

    cmp cx,0

    je call\_rank\_output

    jmp far ptr cal\_final\_inputs

call\_rank\_output:

    call rank\_output

    ret

ra\_nk endp

rank\_output proc near

    mov bx,offset data\_s

    mov cx,5

    mov si,0

    mov di,0

    shuchu data\_s

ran\_output\_s0:

    push cx

    mov cx,12

    push si

ran\_output\_s1:

    mov al,ds:[bx+si]

    mov bx,offset after\_cal

    mov ds:[bx+di+2],al      ;姓名字符串复制

    mov bx,offset data\_s

    inc si

    inc di

    loop ran\_output\_s1

    pop si          ;将下面需要的si拿出来

     ;成绩排序后输出

    pop cx

    mov al,cl

    add al,30h

    mov bx,offset after\_cal

    mov ds:[bx+23],al

    push cx

    push si

    call rank\_last\_output

    pop si

    pop cx

    add si,105           ;将指针指到下一行，+105

    mov di,0

    loop ran\_output\_s0

    ret

rank\_output endp

rank\_last\_output proc near

    mov bx,offset data\_s

    mov di,offset after\_cal

ran\_output\_s2:

    mov al,ds:[bx+si+92]       ;平时成绩

    mov ds:[di+38],al

    mov al,ds:[bx+si+93]

    mov ds:[di+39],al

    mov al,ds:[bx+si+94]

    mov ds:[di+40],al

    mov al,ds:[bx+si+96]       ;大作业成绩

    mov ds:[di+55],al

    mov al,ds:[bx+si+97]

    mov ds:[di+56],al

    mov al,ds:[bx+si+98]

    mov ds:[di+57],al

    mov al,ds:[bx+si+100]       ;最终成绩

    mov ds:[di+73],al

    mov al,ds:[bx+si+101]

    mov ds:[di+74],al

    mov al,ds:[bx+si+102]

    mov ds:[di+75],al

    shuchu after\_cal

    ret

rank\_last\_output endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;搜索查找子程序

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

search proc near

    mov ax,data

    mov ds,ax

sear\_again:

    shuchu search\_prompt

    shuru

    cmp al,49       ;al==1

    je sear\_by\_name

    cmp al,50       ;al==2

    je sear\_by\_id

    shuchu WARNING

    ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;用名字搜索

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sear\_by\_name:

    shuchu name\_prompt

    mov cx,12

    mov bx,offset student

    mov si,0

n\_in\_search:

    shuru

    cmp al,0dh              ;检查回车

    je n\_ss\_search

    mov ds:[bx+si],al       ;先将输入的字符串存好

    inc si

    loop n\_in\_search

n\_ss\_search:

    mov cx,5                ;一共5行数据

    mov si,0                ;si记得清零！

    mov bx,offset student

    mov bp,offset data\_s

n\_s1\_search:

    push cx

    mov cx,12               ;姓名12个空间

n\_s2\_search:

    mov ah,ds:[bx+si]

    mov al,ds:[bp+si]

    cmp ah,al

    jne n\_s3\_search                ;相等就继续比较,不相等就跳走

    inc si

    cmp si,12               ;是不是12个字符都准确

    je output\_student

    loop n\_s2\_search

n\_s3\_search:

    mov si,0               ;到下一行开始

    add bp,105

    pop cx

    loop n\_s1\_search

    shuchu hint5

    jmp far ptr sear\_again

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;用id搜索

sear\_by\_id:

    shuchu id\_prompt

    mov cx,11               ;学号10个位

    mov bx,offset id

    mov si,0

id\_in\_search:

    shuru

    cmp al,0dh              ;检查回车

    je id\_ss\_search

    mov ds:[bx+si],al       ;先将输入的字符串存好

    inc si

    loop id\_in\_search

id\_ss\_search:

    mov cx,5                ;一共5行数据

    mov si,0                ;si记得清零！

    mov bx,offset id

    mov bp,offset data\_s

id\_s1\_search:

    push cx

    mov cx,10               ;学号10个空间

id\_s2\_search:

    mov ah,ds:[bx+si]

    mov al,ds:[bp+si+13]      ;学号字符开始的位置为13

    cmp ah,al

    jne id\_s3\_search                 ;相等就继续比较,不相等就跳走

    inc si

    cmp si,10               ;是不是10个字符都准确

    je output\_student

    loop id\_s2\_search

id\_s3\_search:

    mov si,0               ;到下一行开始

    add bp,105

    pop cx

    loop id\_s1\_search

    shuchu hint5

    jmp far ptr sear\_again

;最后找到正确的学生信息并输出

output\_student:

    shuchu hint2    ;shuchuaccu data\_s

    mov ah,2

    mov dl,13

    int 21h         ;补一个回车

    mov dl,10

    int 21h         ;补一个换行

    mov cx,104      ;每一行105个字符,0-104

    mov si,0

output\_student\_s:

    mov ah,2

    mov dl,ds:[bp+si]

    int 21h

    inc si

    loop output\_student\_s

    jmp far ptr choice

search endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;成绩计算子程序

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cal\_culate proc near

    cal:

; 计算总成绩

    mov ax,0

    mov di,offset lastgrade

    mov ds:[di],ax

    mov di,offset ave\_total

    mov ds:[di],ax      ;开始前先将对应的存储空间清零

    mov si,0

    mov cx, 16

calc\_avg:

    ;计算总数

    push cx

    mov ax,0

    mov al,ds:[bx+si+24]   ;百位，平时成绩的存储偏移地址为每一行的第24个字节位

    sub al,30h          ;将字符串转换为数字

    mov dl,100      ;百位的数乘100

    mul dl

    add ds:[di],ax      ;存储

    inc si

    mov al,ds:[bx+si+24]   ;十位

    sub al,30h          ;将字符串转换为数字

    mov dl,10       ;十位的数乘10

    mul dl

    add ds:[di],ax      ;存储

    inc si

    mov al,ds:[bx+si+24]   ;个位

    sub al,30h          ;将字符串转换为数字

    add ds:[di],ax      ;存储

    inc si

    inc si          ;每个成绩之后的空格

    pop cx

    loop calc\_avg

    ;存储平均数

    mov ax, data

    mov ds,ax

    mov dx,0            ;清零防止后面出错

    mov al,ds:[di]      ;低位在前，高位在后

    mov ah,ds:[di+1]    ;将平均数算平均前的总数传给ax

    mov cx,4            ;设置右移位数

    shr ax,cl            ;将ax寄存器中的值向右移动4位，相当于除以16

    push ax

    push di

    ;将平时成绩乘以0.4,先将其数据处理并加入到最终成绩计算区域中，再进行字符串输出

    mov dx,4

    mul dl

    mov dx,10

    div dl

    mov di,offset total

    mov ds:[di],al

    pop di

    pop ax

;平时成绩处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;十六进制转为十进制——平时作业

    cmp al,64h          ;如果平均分=100分

    je one\_hundred

    ;将平时成绩乘以0.4

    ;mov dx,10

    ;mul dl

    ;mov dx,40

    ;div dl

    ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    push ax             ;输出一个0在前面

    mov al,'0'

    mov ds:[bx+92],al    ;将字符保存

    pop ax

sixteen\_to\_ten:

    ;push ax

    ;shl ax,1            ;把ax寄存器左移一位，用ah拿到16进制的高位

    mov dx,0

    mov si,2            ;存余数由个位存起

    mov cx,2

one\_or\_zero:

    mov ah,0            ;每次进来余数位先清零,用上次的商去除

    mov dl,10           ;将得到的数除以10

    div dl

    add ah,30h          ;将余数转换为字符

    mov ds:[bx+si+92],ah    ;将字符保存

    dec si

    loop one\_or\_zero

    jmp last\_score\_turning

one\_hundred:        ;平时成绩为100分的情况

   ;默认平时成绩40分

   mov ax,data

   mov ds,ax

   ;mov bx,offset data\_s

   mov di,92        ;平时成绩存放处

   mov al,'1'

   mov ds:[bx+di],al

   mov al,'0'

   mov ds:[bx+di+1],al

   mov al,'0'

   mov ds:[bx+di+2],al

   ;mov bx,offset data\_s    ;每次外循环需要重置bx，指向第一个存储位置

;大作业成绩处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

last\_score\_turning:

    ;将字符数字变成实数并存入缓冲区

    mov di,offset lastgrade

    mov si,0

    mov ax,0

    mov ds:[si],ax          ;先对缓冲区清零;;;;;;;;;;;;;;;;;;;;;;;;;;

    mov al,ds:[bx+si+88]   ;百位,88为大作业存储百位数的偏移地址

    sub al,30h          ;将字符串转换为数字

    mov dl,100      ;百位的数乘100

    mul dl

    add ds:[di],ax      ;存储

    inc si

    mov al,ds:[bx+si+88]   ;十位

    sub al,30h          ;将字符串转换为数字

    mov dl,10       ;十位的数乘10

    mul dl

    add ds:[di],ax      ;存储

    inc si

    mov al,ds:[bx+si+88]   ;个位

    sub al,30h          ;将字符串转换为数字

    add ds:[di],ax      ;存储

    mov al,ds:[di]

    mov ah,0

    push ax

    push di

    ;将大作业成绩乘以0.6,先将其数据处理并加入到最终成绩计算区域中，再进行字符串输出

    mov dx,6

    mul dl

    mov dx,10

    div dl

    mov di,offset total

    add ds:[di],al

    pop di

    pop ax

    ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

    cmp al,64h          ;如果平均分=100分

    je las\_one\_hundred

    mov al,ds:[bx+88]

    mov ds:[bx+96],al   ;第1个字符

    mov al,ds:[bx+89]

    mov ds:[bx+97],al   ;第2个字符

    mov al,ds:[bx+90]

    mov ds:[bx+98],al   ;第3个字符

    jmp final\_score\_turning

las\_one\_hundred:        ;大作业成绩为100分的情况

    ;默认大作业满分60分

    mov ax,data

    mov ds,ax

    mov di,96        ;大作业成绩存放处

    mov al,'1'

    mov ds:[bx+di],al

    mov al,'0'

    mov ds:[bx+di+1],al

    mov al,'0'

    mov ds:[bx+di+2],al

;大期末总分成绩处理\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

final\_score\_turning:

    mov di,offset total

    mov al,ds:[di]

    ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;十六进制转为十进制——期末总分

    cmp al,64h          ;如果平均分=100分

    je final\_one\_hundred

    push ax             ;输出一个0在前面

    mov al,'0'

    mov ds:[bx+100],al    ;将字符保存

    pop ax

    mov dx,0

    mov si,2            ;存余数由个位存起

    mov cx,2

final\_one\_or\_zero:

    mov ah,0            ;每次进来余数位先清零,用上次的商去除

    mov dl,10           ;将得到的数除以10

    div dl

    add ah,30h          ;将余数转换为字符

    mov ds:[bx+si+100],ah    ;将字符保存

    dec si

    loop final\_one\_or\_zero

    ret

final\_one\_hundred:        ;期末总成绩为100分的情况

    mov ax,data

    mov ds,ax

    mov di,100        ;最终成绩存放处

    mov al,'1'

    mov ds:[bx+di],al

    mov al,'0'

    mov ds:[bx+di+1],al

    mov al,'0'

    mov ds:[bx+di+2],al

    ret

cal\_culate endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;期末成绩评级

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

grade\_levels proc near

;期末成绩评级

;将期末成绩的字符串转化为数字并比较

    mov bx,offset data\_s

    mov di,offset student\_final\_score

    mov cx,5

grade\_levs:

    mov si,0

    mov ax,0

    mov al,ds:[bx+si+100]   ;百位,期末成绩存储百位数的偏移地址

    sub al,30h          ;将字符串转换为数字

    mov dl,100      ;百位的数乘100

    mul dl

    add ds:[di],al      ;存储

    inc si

    mov al,ds:[bx+si+100]   ;十位

    sub al,30h          ;将字符串转换为数字

    mov dl,10       ;十位的数乘10

    mul dl

    add ds:[di],al      ;存储

    inc si

    mov al,ds:[bx+si+100]   ;个位

    sub al,30h          ;将字符串转换为数字

    add ds:[di],al      ;存储

    inc di

    add bx,105      ;下一行

    loop grade\_levs

    mov si,offset student\_level

    mov ah,1

    mov di,offset student\_final\_score

    mov cx,5

    ;换数字完毕，开始比较

lev\_1:

    mov al,ds:[di]

    cmp al,90       ;100-90

    ja lev\_1\_add

lev\_2:

    cmp al,80       ;89-80

    jnb lev\_2\_add

lev\_3:

    cmp al,60       ;79-60

    jnb lev\_3\_add

lev\_4:

    cmp al,60       ;60-

    jb lev\_4\_add

lev\_continue:

    inc di

    loop lev\_1

    jmp lev\_output

lev\_1\_add:

    add ds:[si],ah

    jmp lev\_continue

lev\_2\_add:

    add ds:[si+1],ah

    jmp lev\_continue

lev\_3\_add:

    add ds:[si+2],ah

    jmp lev\_continue

lev\_4\_add:

    add ds:[si+3],ah

    jmp lev\_continue

    ;输出分段个数

lev\_output:

    mov al,30h

    add ds:[si],al

    mov ah,ds:[si]      ;将字符串移动

    inc si

    lea di,lev1

    mov ds:[di+16],ah

    add ds:[si],al

    mov ah,ds:[si]      ;将字符串移动

    inc si

    lea di,lev2

    mov ds:[di+16],ah

    add ds:[si],al

    mov ah,ds:[si]      ;将字符串移动

    inc si

    lea di,lev3

    mov ds:[di+16],ah

    add ds:[si],al

    mov ah,ds:[si]      ;将字符串移动

    inc si

    lea di,lev4

    mov ds:[di+16],ah

    shuchu lev1

    shuchu lev2

    shuchu lev3

    shuchu lev4

    mov si,offset student\_level

    mov di,offset student\_final\_score

    mov al,0        ;输出后便重置

    mov cx,5

level\_reset:

    mov ds:[si],al

    mov ds:[di],al

    inc si

    inc di

    loop level\_reset

    ret

grade\_levels endp

average\_and\_hilo proc near

    ;平时成绩

    ;将字符数字变成实数并存入缓冲区

    shuchu huiche

    mov bx,offset data\_s

    mov cx,5

av\_and\_hilo\_s0:

    mov di,offset average\_daily

    mov si,0

    mov ax,0

    mov al,ds:[bx+si+92]   ;百位,92为平时作业存储百位数的偏移地址

    sub al,30h          ;将字符串转换为数字

    mov dl,100      ;百位的数乘100

    mul dl

    add ds:[di],ax      ;存储

    inc si

    mov al,ds:[bx+si+92]   ;十位

    sub al,30h          ;将字符串转换为数字

    mov dl,10       ;十位的数乘10

    mul dl

    add ds:[di],ax      ;存储

    inc si

    mov al,ds:[bx+si+92]   ;个位

    sub al,30h          ;将字符串转换为数字

    add ds:[di],ax      ;存储

    add bx,105

    loop av\_and\_hilo\_s0

    mov al,ds:[di]      ;低位在前，高位在后

    mov ah,ds:[di+1]    ;将平均数算平均前的总数传给ax

    mov bx,offset av\_da

    mov dl,5

    div dl

    push ax             ;输出一个0在前面

    mov al,'0'

    mov ds:[bx+24],al    ;将字符保存

    pop ax

    mov dx,0

    mov si,2            ;存余数由个位存起

    mov cx,2

daily\_av\_oz:

    mov ah,0            ;每次进来余数位先清零,用上次的商去除

    mov dl,10           ;将得到的数除以10

    div dl

    add ah,30h          ;将余数转换为字符

    mov ds:[bx+si+24],ah    ;将字符保存

    dec si

    loop daily\_av\_oz

    shuchu av\_da

    mov al,0

    mov ds:[bx+24],al   ;清零

    mov ds:[bx+25],al

    mov ds:[bx+26],al

    mov ax,0

    mov ds:[di],al

    mov ds:[di+1],al

    mov bx,offset data\_s

    mov cx,5

av\_and\_hilo\_s1:

    ;大作业成绩

    ;将字符数字变成实数并存入缓冲区

    mov di,offset average\_last

    mov si,0

    mov ax,0

    mov al,ds:[bx+si+96]   ;百位,96为大作业存储百位数的偏移地址

    sub al,30h          ;将字符串转换为数字

    mov dl,100      ;百位的数乘100

    mul dl

    add ds:[di],ax      ;存储

    inc si

    mov al,ds:[bx+si+96]   ;十位

    sub al,30h          ;将字符串转换为数字

    mov dl,10       ;十位的数乘10

    mul dl

    add ds:[di],ax      ;存储

    inc si

    mov al,ds:[bx+si+96]   ;个位

    sub al,30h          ;将字符串转换为数字

    add ds:[di],ax      ;存储

    add bx,105

    loop av\_and\_hilo\_s1

    mov al,ds:[di]

    mov ah,ds:[di+1]

    mov bx,offset av\_la

    mov dl,5

    div dl

    push ax             ;输出一个0在前面

    mov al,'0'

    mov ds:[bx+24],al    ;将字符保存

    pop ax

    mov dx,0

    mov si,2            ;存余数由个位存起

    mov cx,2

last\_av\_oz:

    mov ah,0            ;每次进来余数位先清零,用上次的商去除

    mov dl,10           ;将得到的数除以10

    div dl

    add ah,30h          ;将余数转换为字符

    mov ds:[bx+si+24],ah    ;将字符保存

    dec si

    loop last\_av\_oz

    shuchu av\_la

    mov al,0

    mov ds:[bx+24],al

    mov ds:[bx+25],al

    mov ds:[bx+26],al

    mov ax,0

    mov ds:[di],al

    mov ds:[di+1],al

    mov bx,offset data\_s

    mov cx,5

av\_and\_hilo\_s2:

    ;大作业成绩

    ;将字符数字变成实数并存入缓冲区

    mov di,offset average\_final

    mov si,0

    mov ax,0

    mov al,ds:[bx+si+100]

    sub al,30h          ;将字符串转换为数字

    mov dl,100      ;百位的数乘100

    mul dl

    add ds:[di],ax      ;存储

inc si

    mov al,ds:[bx+si+100]   ;十位

    sub al,30h          ;将字符串转换为数字

    mov dl,10       ;十位的数乘10

    mul dl

    add ds:[di],ax      ;存储

inc si

    mov al,ds:[bx+si+100]   ;个位

    sub al,30h          ;将字符串转换为数字

    add ds:[di],ax      ;存储

    add bx,105

loop av\_and\_hilo\_s2

    mov al,ds:[di]

mov ah,ds:[di+1]

    mov bx,offset av\_fi

    mov dl,5

div dl

    push ax             ;输出一个0在前面

    mov al,'0'

    mov ds:[bx+24],al    ;将字符保存

pop ax

    mov dx,0

    mov si,2            ;存余数由个位存起

    mov cx,2

final\_av\_oz:

    mov ah,0            ;每次进来余数位先清零,用上次的商去除

    mov dl,10           ;将得到的数除以10

div dl

    add ah,30h          ;将余数转换为字符

    mov ds:[bx+si+24],ah    ;将字符保存

    dec si

loop final\_av\_oz

    shuchu av\_fi

    mov al,20h

    mov ds:[bx+24],al

    mov ds:[bx+25],al

    mov ds:[bx+26],al

    mov ax,0

    mov ds:[di],al

    mov ds:[di+1],al

    mov bx,offset data\_s

    mov bp,offset low\_score

    ;最高最低输出

    mov si,0

    mov al,ds:[bx+si+92]

    mov ds:[bp+si+16],al

    inc si

    mov al,ds:[bx+si+92]

    mov ds:[bp+si+16],al

    inc si

    mov al,ds:[bx+si+92]

    mov ds:[bp+si+16],al

mov si,0

    mov al,ds:[bx+si+96]

    mov ds:[bp+si+25],al

    inc si

    mov al,ds:[bx+si+96]

    mov ds:[bp+si+25],al

    inc si

    mov al,ds:[bx+si+96]

    mov ds:[bp+si+25],al

mov si,0

    mov al,ds:[bx+si+100]

    mov ds:[bp+si+35],al

    inc si

    mov al,ds:[bx+si+100]

    mov ds:[bp+si+35],al

    inc si

    mov al,ds:[bx+si+100]

    mov ds:[bp+si+35],al

    mov si,0

    shuchu low\_score

;最高成绩输出

    shuchu huiche

    shuchu huiche

    mov si,0

    mov bx,offset data\_s

    mov bp,offset high\_score

    add bx,420

    mov al,ds:[bx+si+92]

    mov ds:[bp+si+16],al

    inc si

    mov al,ds:[bx+si+92]

    mov ds:[bp+si+16],al

    inc si

    mov al,ds:[bx+si+92]

    mov ds:[bp+si+16],al

mov si,0

    mov al,ds:[bx+si+96]

    mov ds:[bp+si+25],al

    inc si

    mov al,ds:[bx+si+96]

    mov ds:[bp+si+25],al

    inc si

    mov al,ds:[bx+si+96]

    mov ds:[bp+si+25],al

mov si,0

    mov al,ds:[bx+si+100]

    mov ds:[bp+si+35],al

    inc si

    mov al,ds:[bx+si+100]

    mov ds:[bp+si+35],al

    inc si

    mov al,ds:[bx+si+100]

    mov ds:[bp+si+35],al

    mov si,0

    shuchu high\_score

    ret

average\_and\_hilo endp

;菜单显示子程序

show proc near

    mov ax,data

    mov ds,ax

    mov ax,3  ;清屏

    int 10h

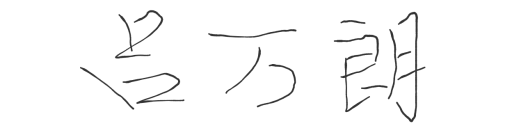
    shuchu LIST

    ret

show endp

code ends

end start

签名： ****

日期： 2023/6/22