**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：陈文笛 学号：2021103285 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

程序框架：

ex1.asm-绘制系统界面

ex2.asm-信息录入

ex3.asm-计算学生的最终成绩

ex4.asm-查询功能

ex5.asm-排序

ex6.asm-分数统计信息显示

ex8.asm-整合后的总系统

定义学生信息使用的数据段：

以result结尾的用于存储对应的十进制数。比如finalresult存储的是final的十进制数。

sname db 255, 0, 255 dup('$'); 学生姓名每人5个字符，每个字符以$结尾。

sid db 255, 0, 255 dup('$'); 学号，5个字符，每个字符串以$结尾。

scores db 255, 0, 255 dup('|');每次的成绩3个字符，每个字符串以$结尾 。

scoresresult db 255,0,255 dup('|') ;每次的成绩的的十进制数字

final db 255, 0, 255 dup('$');加权总成绩的ascii,每个字符串以$结尾

finalresult db 255,0,255 dup('|');加权总成绩的十进制数字

rankresult db 20,10,01,02,03,04,05,06,07,08,09,10,10 dup('|')

srankresult db 255,0,255 dup('|')

srank db 255,0,20 dup('|') ;每个学生排名的ascii

hla4result db 255,0,255 dup('|');hla:high low average 4 segment

hla4 db 255,0,255 dup('|');hla:high low average 4 segment

各模块代码思路：

信息录入

* 对于姓名、学号，要求输入必须为5个字符，不能多也不能少。

采用mov ah,0ah int 21h 进行接收字符串，存放在sname和score。

* 对于17次成绩（作业成绩和大作业成绩），要求输入必须为xxx(类似100、099、098)。

由于使用mov ah, 0ah会重叠在同一行所以使用mov ah,01h进行接收。

对于一次的成绩，使用三次mov ah,01h，一次mov byte ptr ds:[si], '$'。

自动计算学生的最终成绩（以1个学生为例）

1. 将17次成绩的字符串存放的缓冲区。每个字符串以$符号结尾，一个成绩字符串（如‘100$’）占用4个字节。17次成绩的字符串全部存放后以\符号结尾。
2. 采用循环将17次成绩化为数字存入scoresresult数据段。
3. 将16次平时成绩相加后除以16乘以4，将1次大作业成绩相加后乘以6，将两个数字相加并除以10，得到最终成绩。
4. 最终成绩以数字与字符串的形式分别存储

查询功能

1. 询问需要按学号查询还是姓名查询，比对接收的字符，跳转到对应位置
2. （以为姓名查询为例，学号类似）要求输入姓名，将接收到的姓名存入缓冲区。采用repz cmpsb进行姓名字符串的比对。
3. 由于我在输入名字的时候，设置了在最后的名字后面加上“｜”符号，作为所有名字结束的标志。所以，这里使用cmp byte ptr ds:[si],'|'来判断是否还有下个字符串。
4. 若比对完所有字符串，仍为找到，输出没找到。若找到，跳转到信息输出函数。在比对时，用bl记录下该学生是第几个学生，在信息输出函数中要用到。
5. 利用bl记录的学生位数，利用乘法与加法，计算出数据段中该学生的信息的位置，并进行输出。

成绩排序

使用冒泡排序对学生的最终成绩进行排序。

1.设置一个学生编号的数据段。第一个学生为01，以此类推。

rankresult db 20,10,01,02,03,04,05,06,07,08,09,10,10 dup('\')

2.采用冒泡排序对最终成绩进行排序。

* 比较两个成绩的大小，小于则不交换，大于则交换，这样最终成绩会由小到大排列。
* 第1次比较9次（前10个），第2次比较8次（前9个），以此类推。
* 交换成绩的时候，同时也要交换对应的学生编号，因为这样才能在成绩位置改变后仍然知道成绩属于哪一位学生。（虽然数字成绩位置改变了，但是ascii码成绩位置没有变，所以还是可以输出正确的最终成绩）

3.打印。成绩由小到大排序，同样，对应的编号代表着成绩由小到大的同学。利用循环，以及加法和乘法进行位置的计算，输出对应的编号的学生的全部信息。由此，就打印出了成绩由小到大排序的10个同学的信息。

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

hla4result db 20,0,20 dup('|');hla:high low average 4 segment

hla4result:代表hightest,lowest,average,4个分数段的人数，用于存放这7个统计信息的数据。

最高分：

1. 把第1个学生的最终成绩存放进al，与下一个学生x的成绩进行比较。
2. 若al中的数大则再与下一个学生x+1的成绩进行比较。若al中的数小，则将学生x的成绩存入al，再与下一个学生x+1的成绩进行比较。
3. 重复步骤2，直到所有学生的成绩都经过比较。
4. al寄存器中的数即为最高分，存入hla4result

最低分：

1. 把第1个学生的最终成绩存放进al，与下一个学生x的成绩进行比较。
2. 若al中的数小则再与下一个学生x+1的成绩进行比较。若al中的数大，则将学生x的成绩存入al，再与下一个学生x+1的成绩进行比较。
3. 重复步骤2，直到所有学生的成绩都经过比较。
4. al寄存器中的数即为最低分，存入hla4result

平均分：

1. 利用循环，将平均成绩总和存储在bx中。
2. 总平时成绩除以10（10为学生人数）。
3. al保存的商，是平均成绩，存入hla4result

各个总成绩分数段的人数：

1. 首先把hla4result中，准备存入各个总成绩分数段的人数的内存设置为0。
2. 取出一个学生最终成绩，与90比较。
3. 若小于，比较下一个分数段。若大于，把hla4result中对应内存的值加1，跳转，结束该学生成绩的比较。
4. 其他分数段与90-100分数段类似。
5. 执行步骤2、3、4，直到所有学生的成绩都比较过。

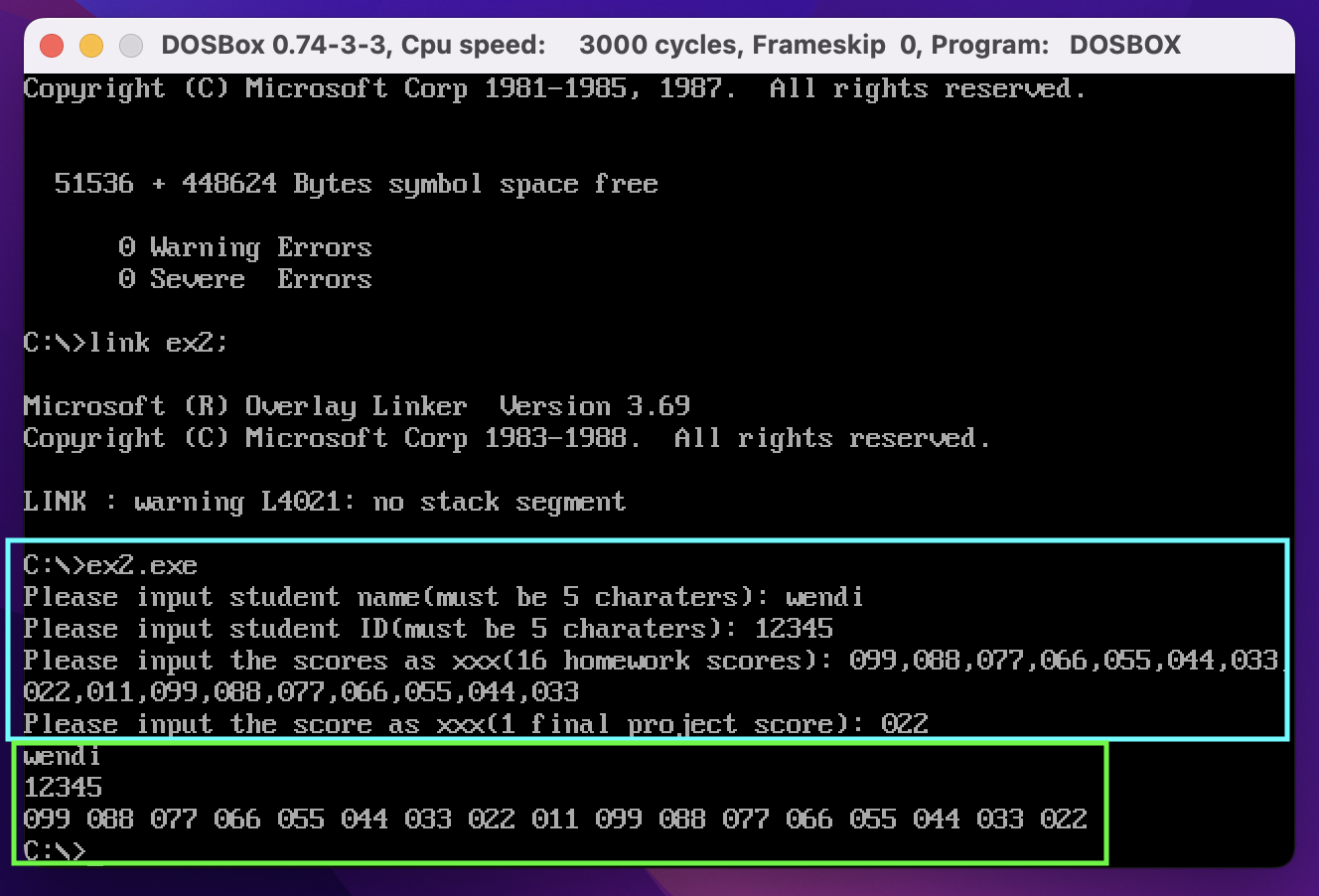
执行完以上函数，hla4result中，将会有7个数，分别代表hightest,lowest,average,4个分数段的人数。

1. 模块功能正确性测试

信息录入：学生的姓名、学号、及16次作业成绩和一次大作业成绩。

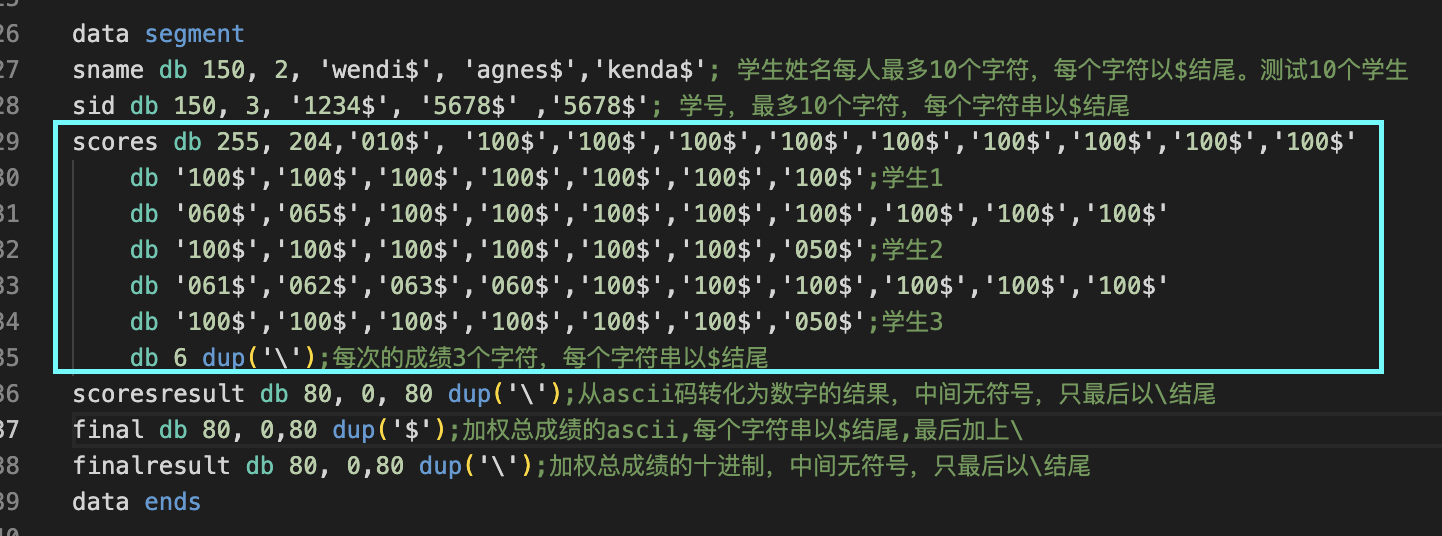
这里展示进行1个学生的信息录入。由于进行多个学生的信息录入，展示时需要用到查询功能，所以多个学生的信息录入正确性测试将在集成测试阶段给出。

蓝框内：信息录入 绿框内：打印录入的信息证明功能正确性

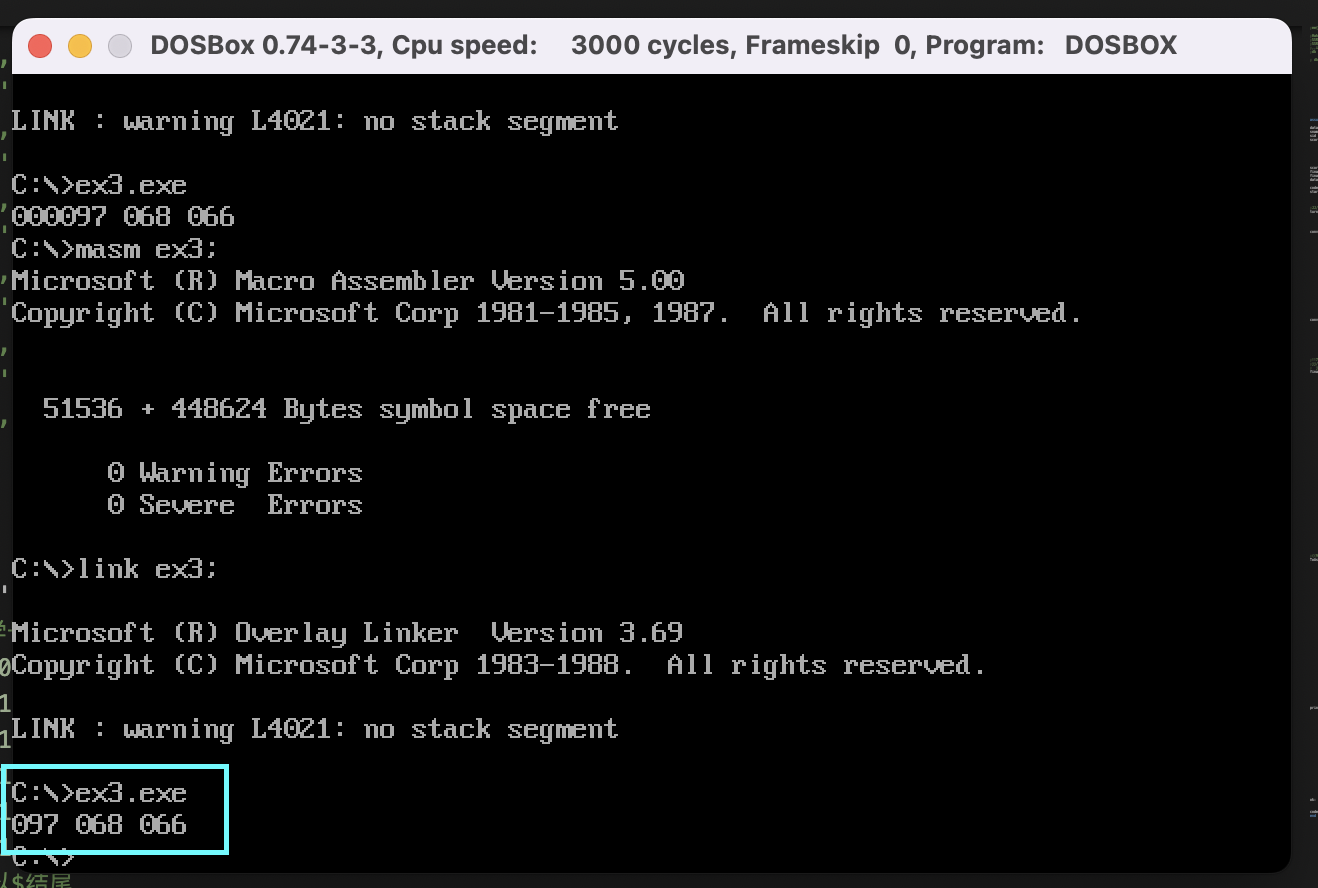


自动计算学生的最终成绩

计算如下图蓝色框中的三个的学生的最终成绩。



三个的学生的最终成绩应为：97.75、68.125、66.15。可见去掉小数部分后与该功能输出的成绩一致。最终成绩功能实现正确。运行结果在蓝框内。

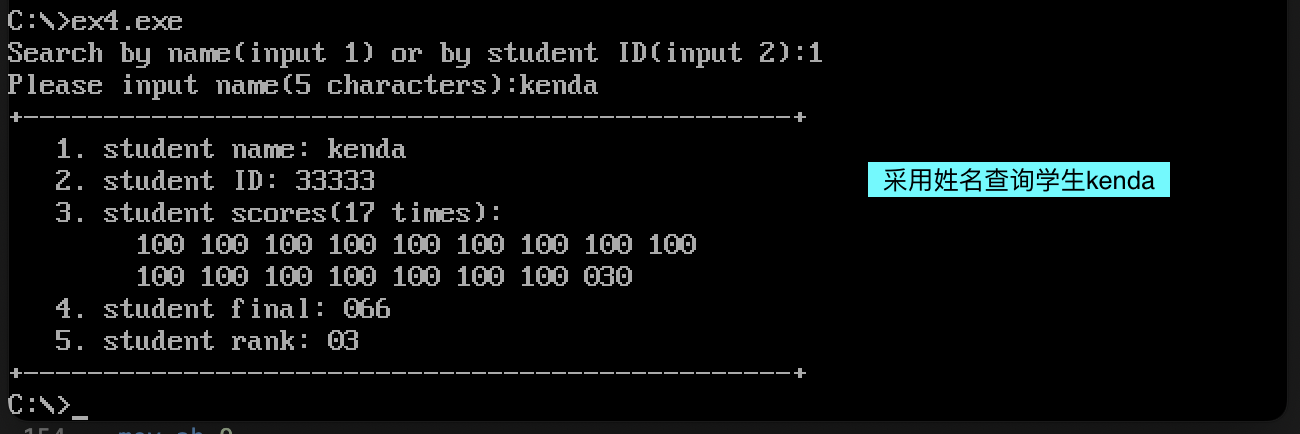
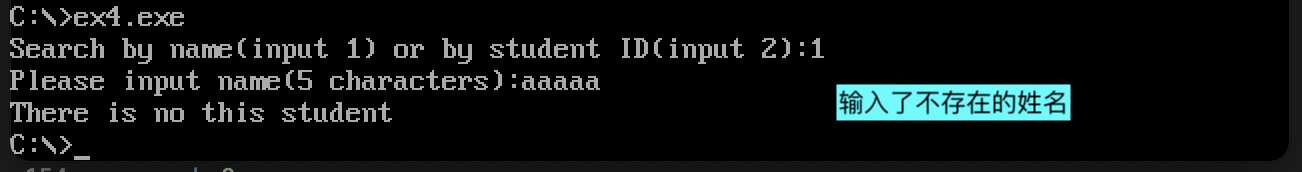
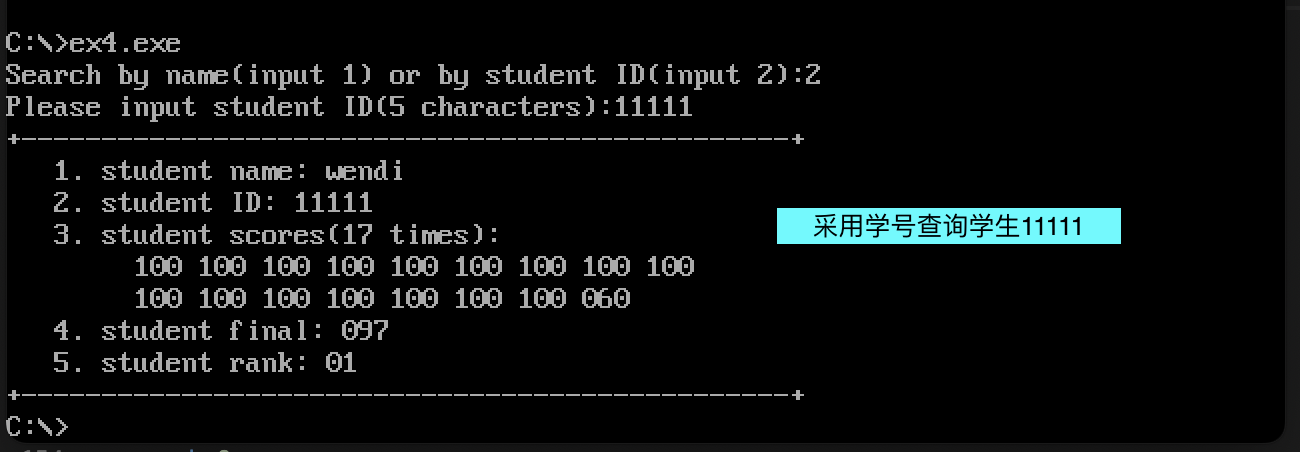
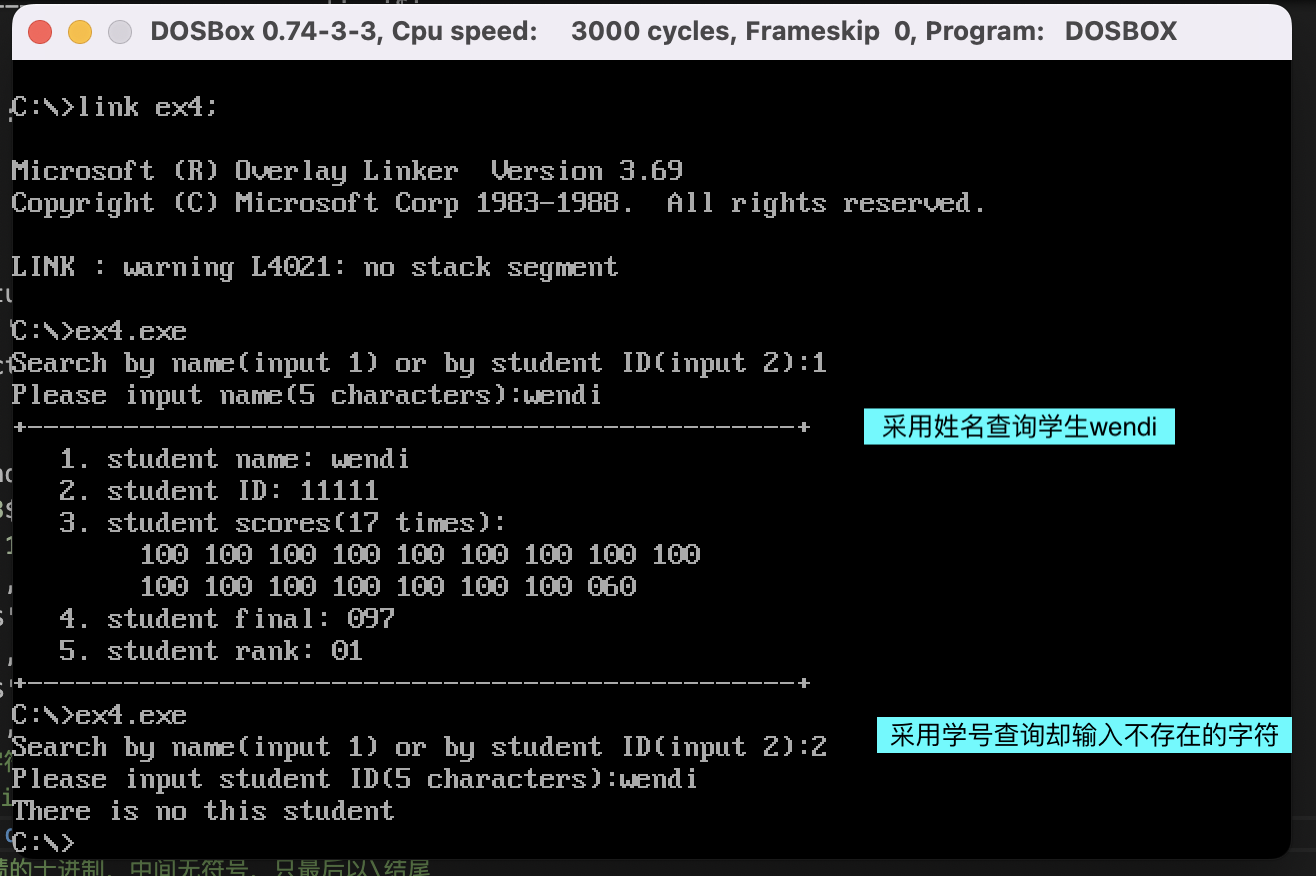


查询功能

数据段保存有三个学生wendi,agnes,kanda的信息，使用查询功能查询某个学生的信息。



通过按姓名查询和学号查询，查询了wendi和kanda的信息，并测试了输入不存在的学生姓名及学号的情况。测试解释在图片中的蓝底框内。

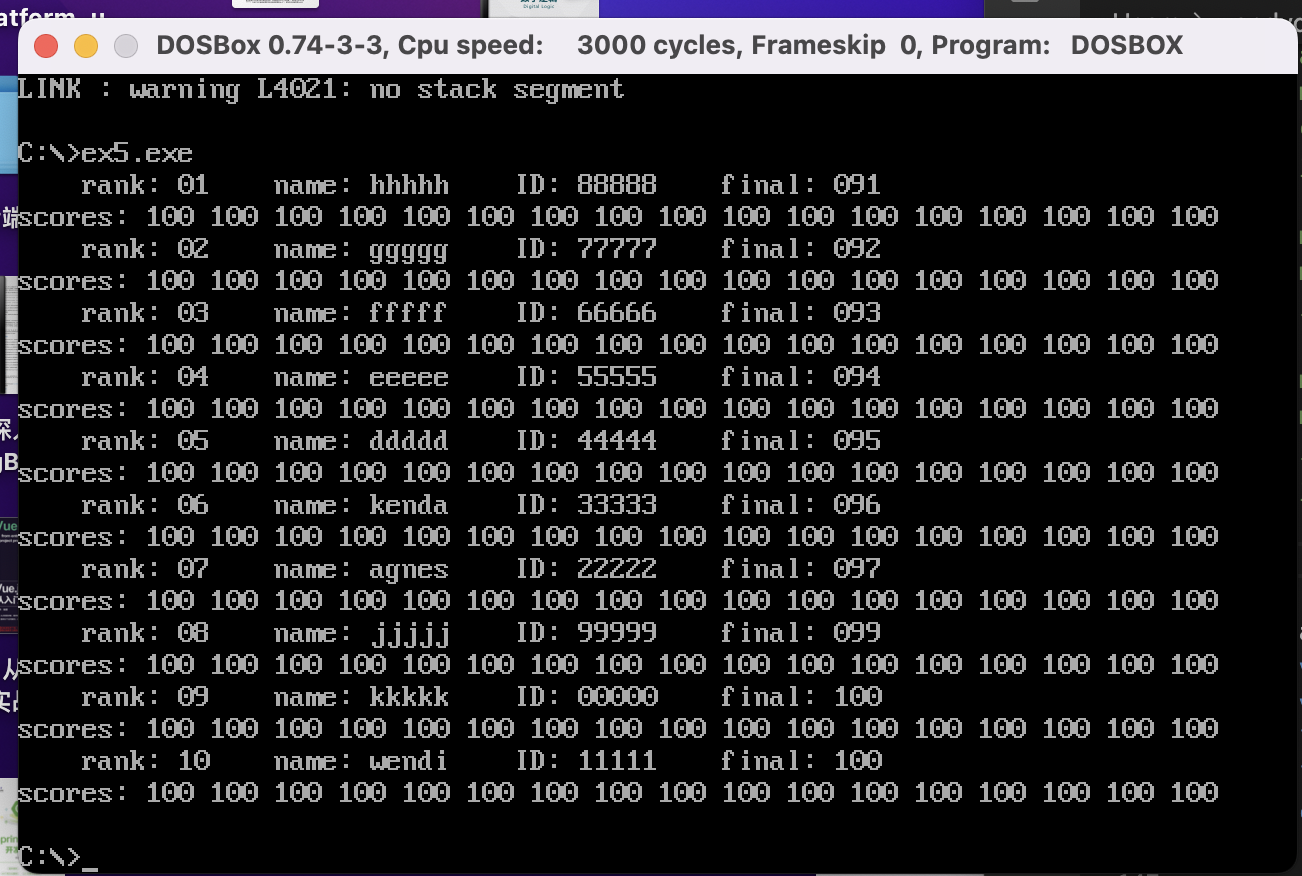


成绩排序

这里还没有整合计算最终成绩的功能，所以最终成绩并不是由平时成绩计算所得。这里的平时成绩只是我为了打印设置的，实际计算排序是由最终成绩（final）决定的。



可以见到，这里第1位同学wendi和最后第10位同学kkkkk，的最终成绩都是100分，所以根据成绩由小到大，他们分别在排序最后两位。而倒数第3位同学hhhhh最终成绩为91分，最低，所以排序在第1位。（平时成绩、大作业成绩都在score内）

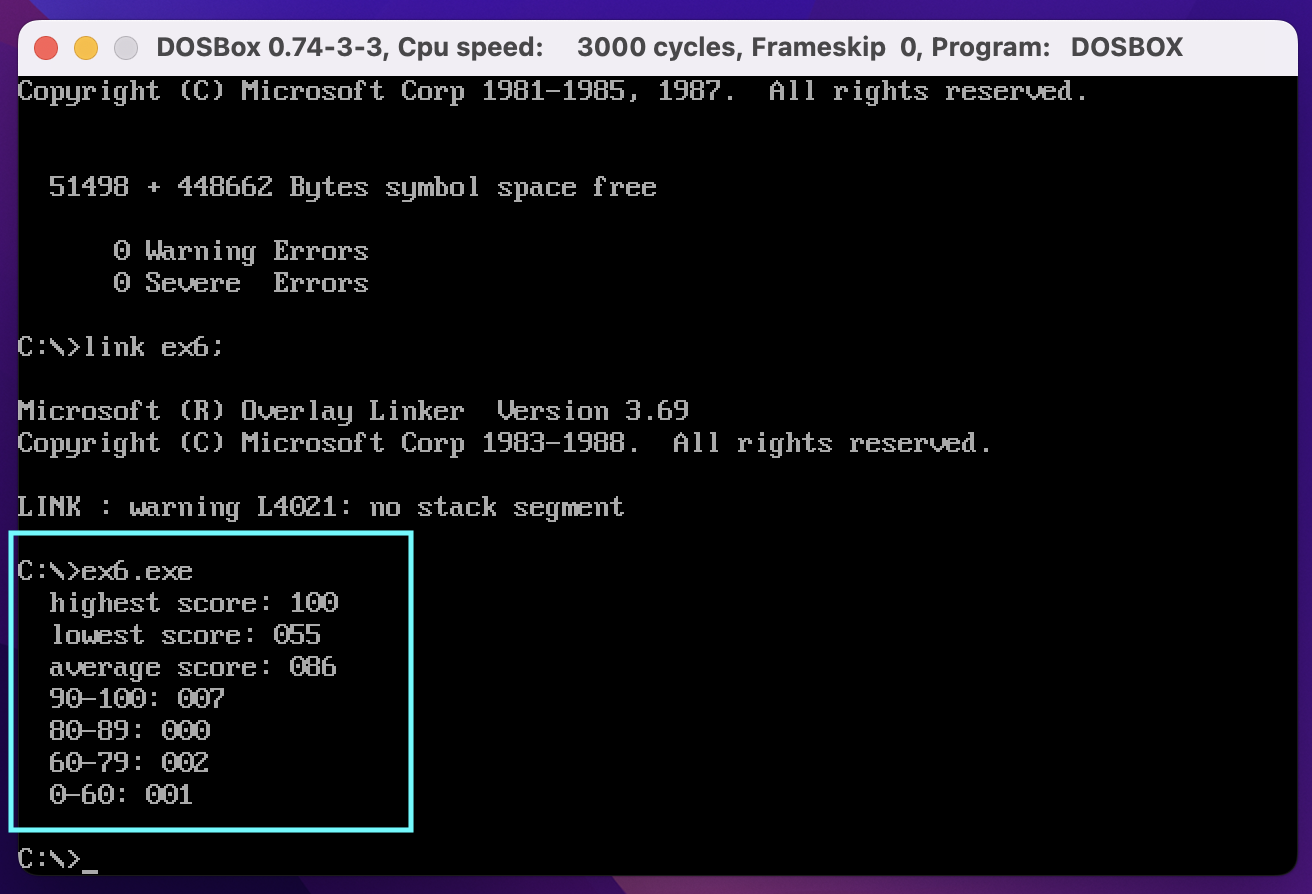


分数统计信息显示：各个总成绩分数段的人数、平均分、最高分、最低分

finalresult db 20,0,100,97,96,95,94,93,92,75,65,55,1 dup('|')

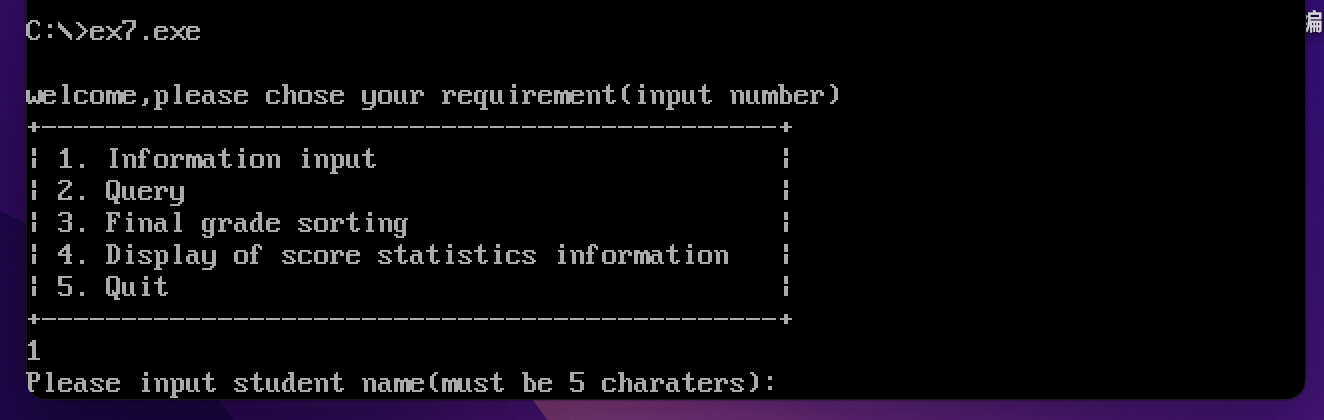
100,97,96,95,94,93,92,75,65,55，为10个学生的成绩。

运行结果在蓝框内。

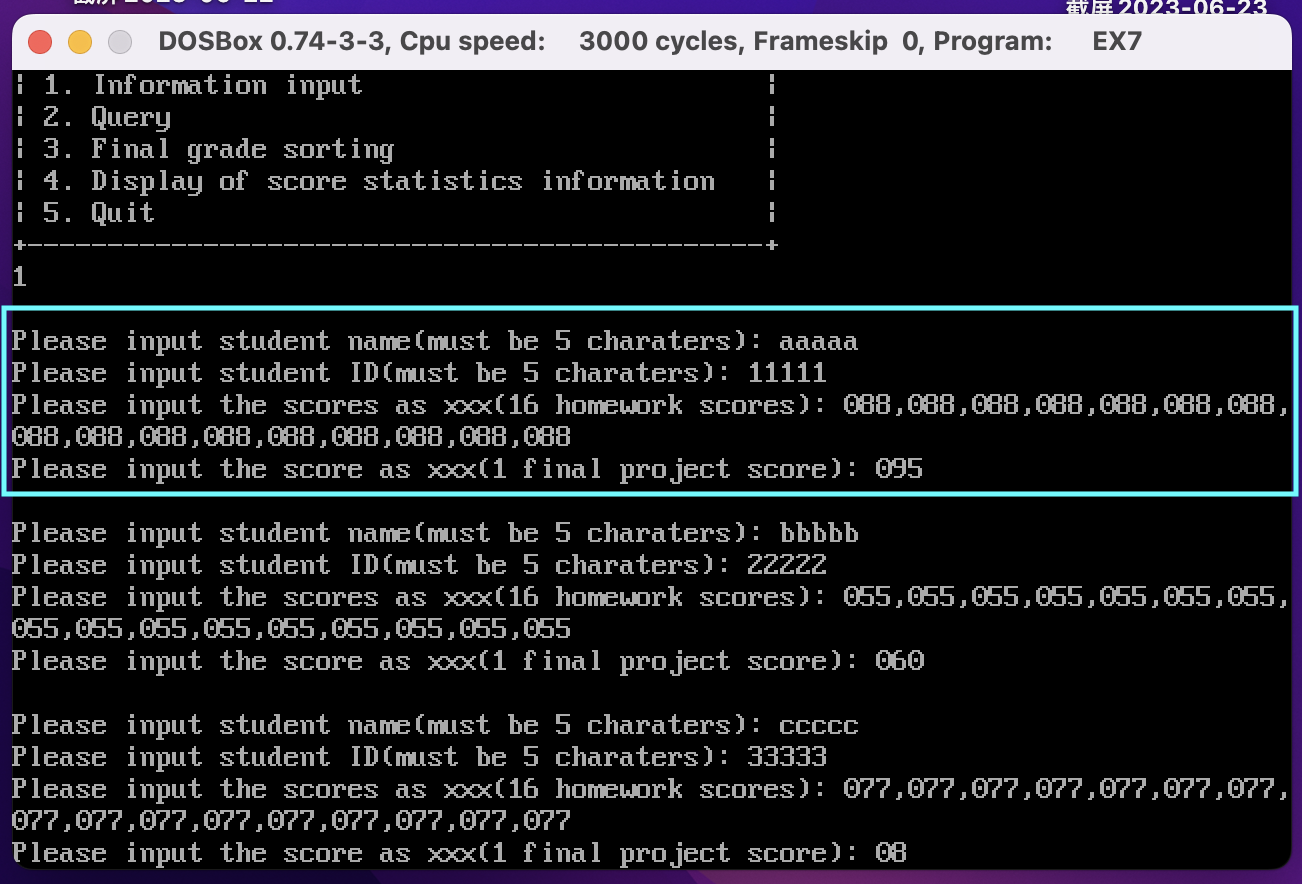


1. 集成测试

进入界面，输入1-5，选择对应的功能。（下面将具体演示）



输入1，进入输入信息功能。

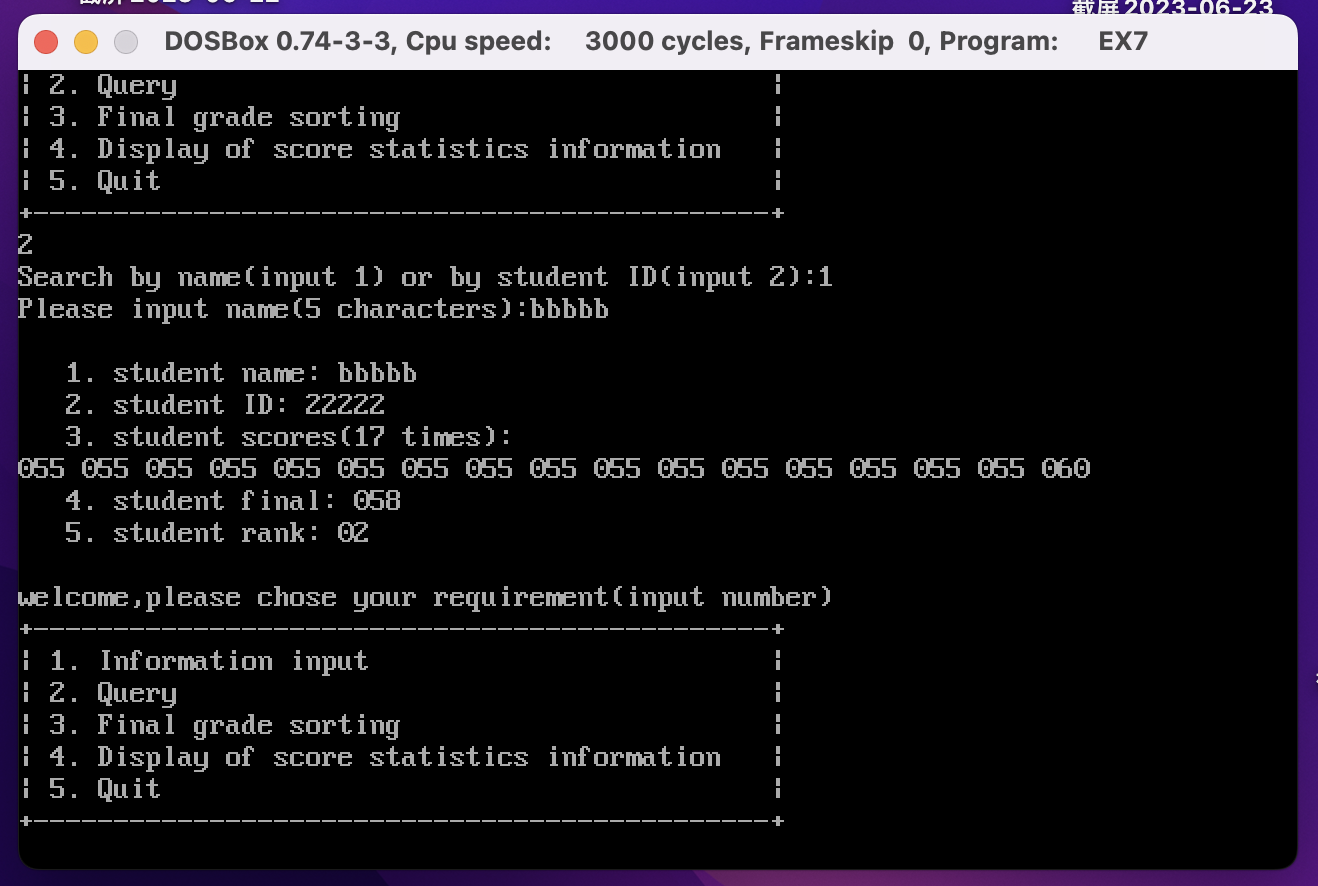
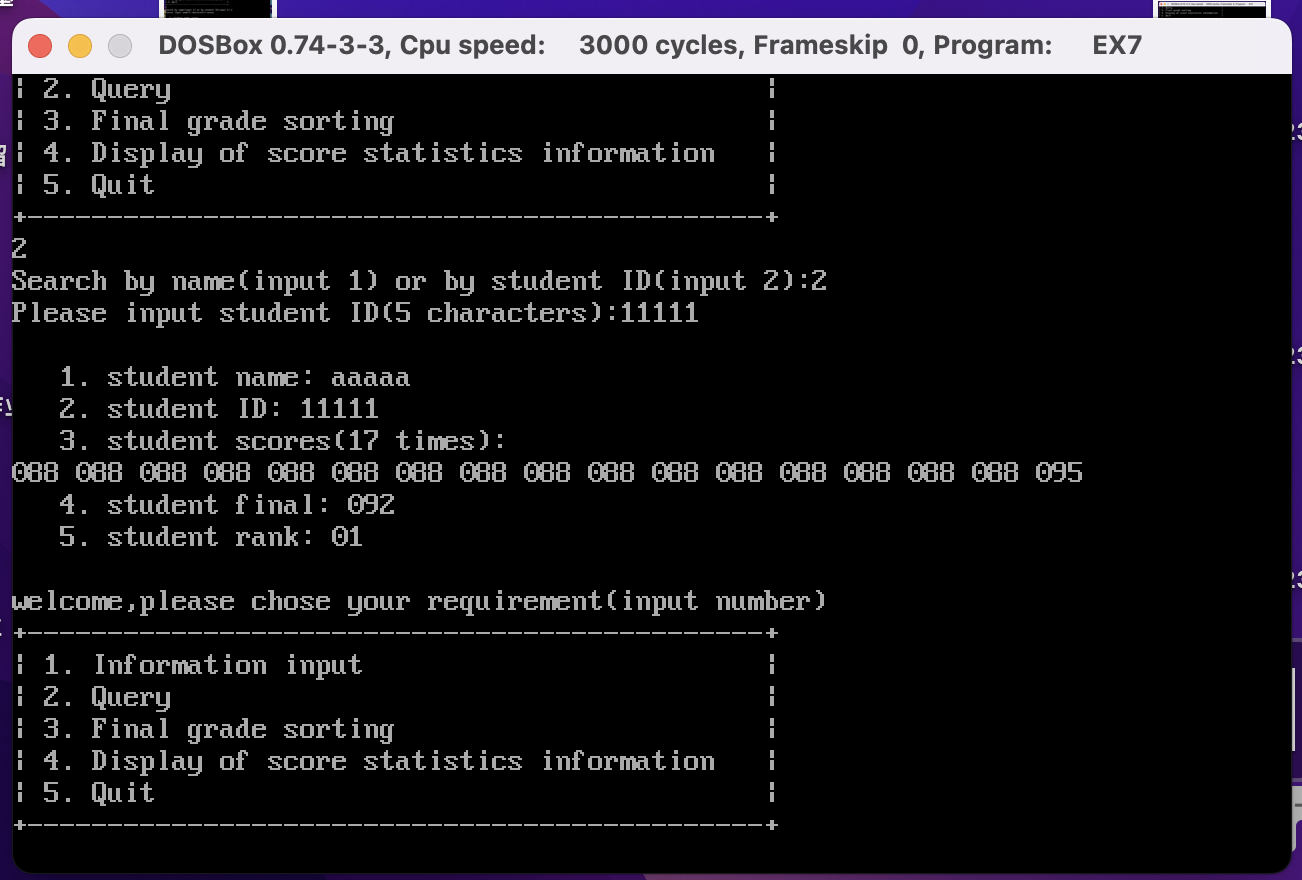


如上图，根据提示输入信息。

输入的限制如下：学生姓名和ID必须等于5个字符、学生成绩必须等于3个字符、必须连续输入10个学生的信息。

输入信息结束后，将重新显示主功能表，可重新选择功能。

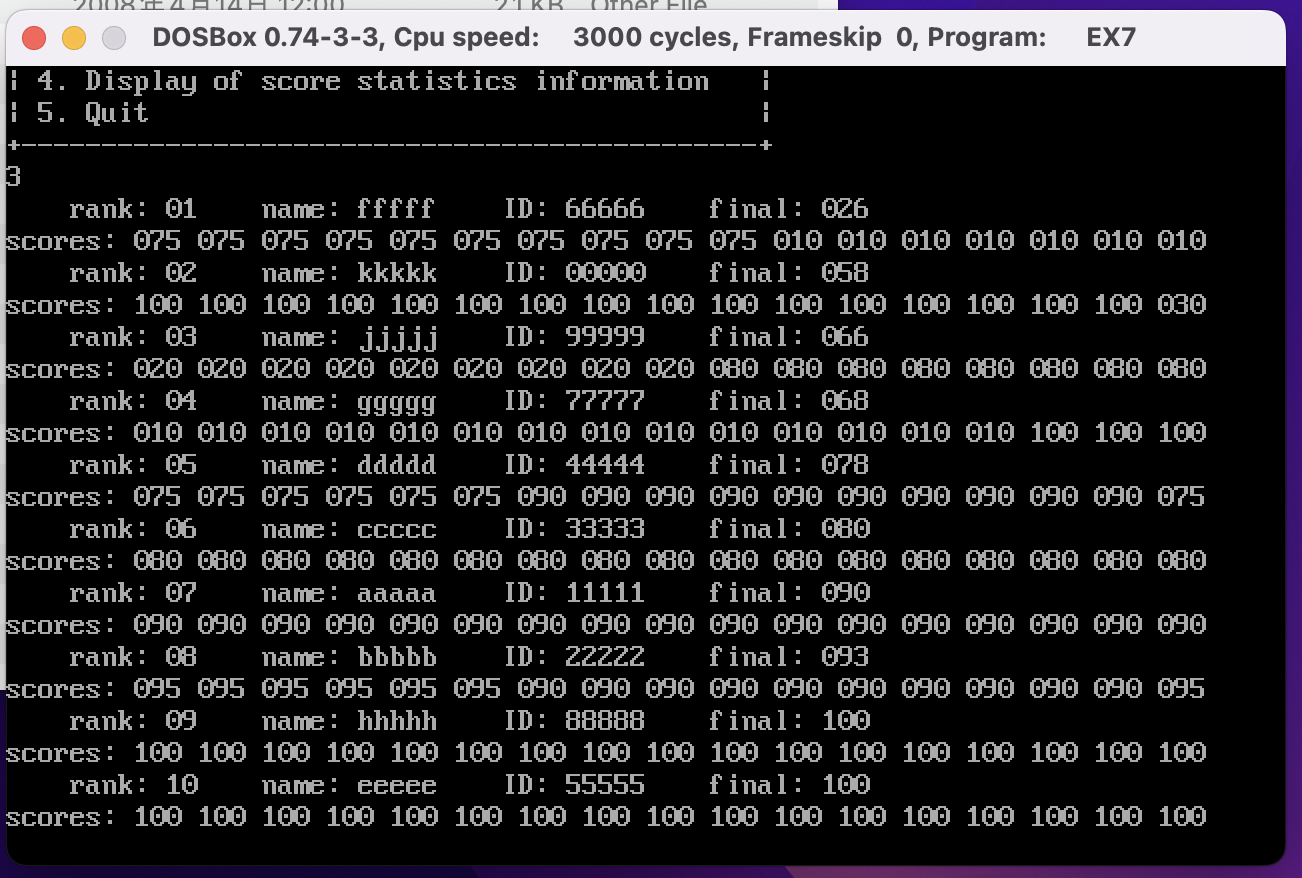
如下图，输入2，进入查询功能。



如上2图，通过学生ID，查询了之前输入的学生aaaa。通过学生姓名，查询了之前输入的学生bbbbb。Student final是系统自动计算的学生最终成绩。

（注：这里的学生排名student rank不是真实的，是为了美观设置的默认排名，进行3.排序操作后，将会显示真实排名）

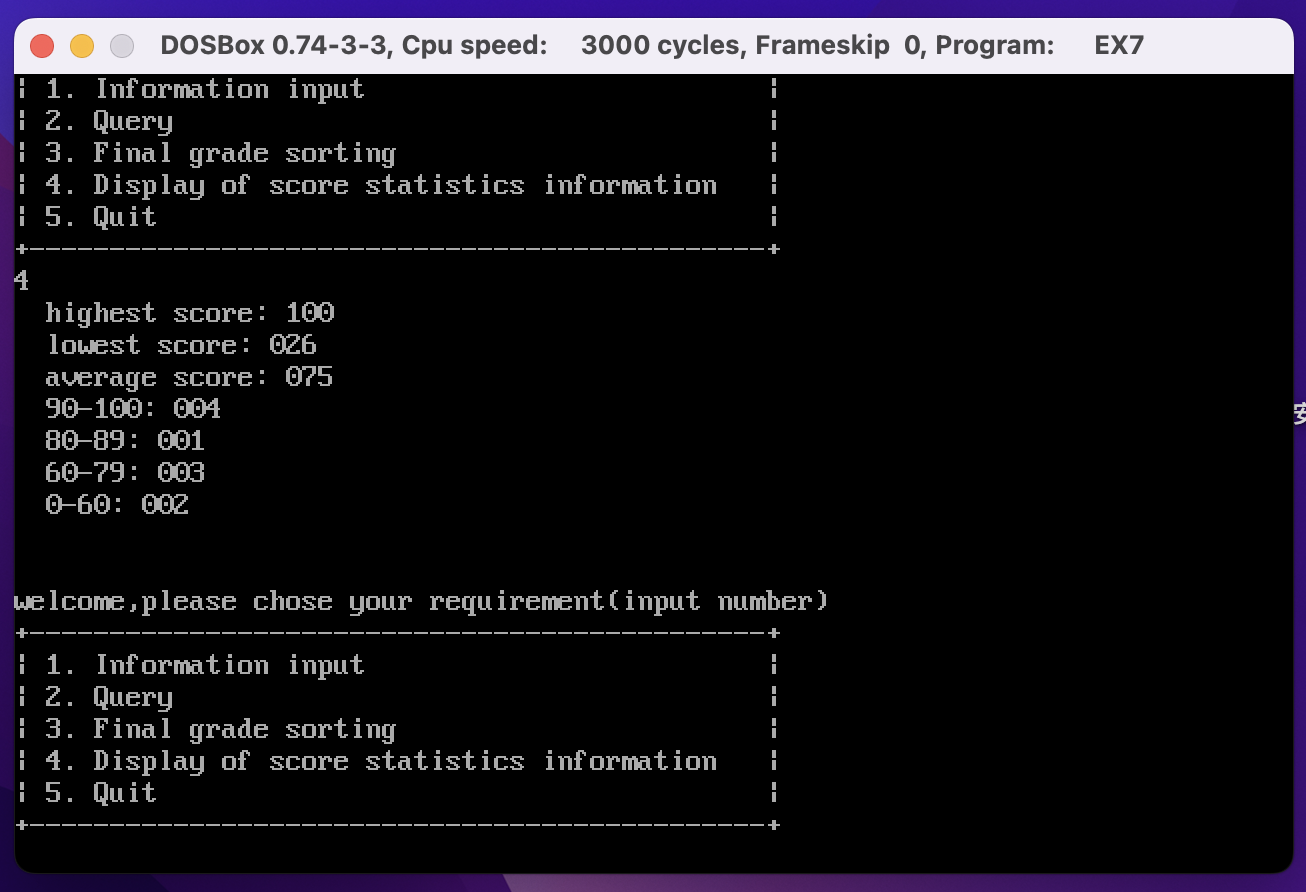
回到功能表，选择3，进行排序。



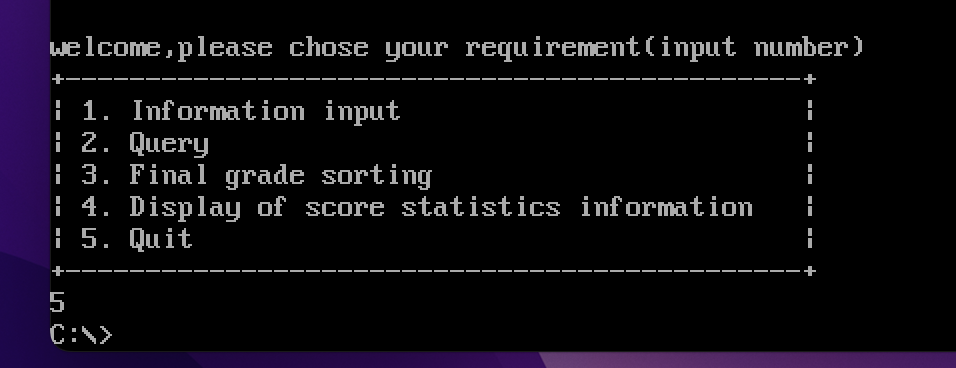
排序结果如上图所示。按任意按键，功能表重新出现。

回到功能表，选择4，进行最终成绩信息统计。

最终成绩：090,093,080,078,100,026,068,100,066,058，统计如下。



回到功能表，选择5，退出系统。如下图。



1. 系统的缺陷分析

* 学生人数限制：系统要求输入10个学生的信息，并且只能处理这10个学生的信息（不能多于或少于）。
* 输入限制：系统要求输入的学生姓名和ID必须等于5个字符、学生成绩必须等于3个字符、必须连续输入10个学生的信息。成绩输入时，按键按下后，无法更改（无法通过删除键进行删除）。
* 代码复用性低：系统中存在一些类似代码重复的情况，例如排序的打印与查询的打印，以及两位十进制转Ascii码与三位十进制转Ascii码的转换。这些函数非常相似又稍有不同，需要进行代码重构以提高代码复用性。

1. 源代码

整合后系统代码

assume cs:code,ds:data

data segment

;用于主界面显示

T DB 'welcome,please chose your requirement(input number)', '$'

LINE DB '+----------------------------------------------+', '$'

LABEL1 DB '| 1. Information input |', '$'

LABEL2 DB '| 2. Query |', '$'

LABEL3 DB '| 3. Final grade sorting |', '$'

LABEL4 DB '| 4. Display of score statistics information |', '$'

LABEL5 DB '| 5. Quit |', '$'

LINE2 DB '+----------------------------------------------+', '$'

;用于信息录入程序

msg1 db 'Please input student name(must be 5 charaters): $'

msg2 db 'Please input student ID(must be 5 charaters): $'

msg3 db 'Please input the scores as xxx(16 homework scores): $'

msg4 db 'Please input the score as xxx(1 final project score): $'

;用于查询程序

LABEL10 DB ' 1. student name: ', '$'

LABEL6 DB ' 2. student ID: ', '$'

LABEL7 DB ' 3. student scores(17 times): ', '$'

LABEL8 DB ' 4. student final: ', '$'

LABEL9 DB ' 5. student rank: ', '$'

amsg1 db 'Search by name(input 1) or by student ID(input 2):', '$'

amsg2 db 'Please input name(5 characters):', '$'

amsg3 db 'Please input student ID(5 characters):', '$'

amsg4 db 'There is no this student', '$'

buf db 20,0,20 dup('$');缓存

;用于排序打印

LABEL11 DB ' name: ', '$'

LABEL12 DB ' ID: ', '$'

LABEL13 DB 'scores: ', '$'

LABEL14 DB ' final: ', '$'

LABEL15 DB ' rank: ', '$'

BLANK DB ' ', '$'

;用于分数统计打印

HIGHTEST DB ' highest score: ', '$'

LOWEST DB ' lowest score: ', '$'

AVERAGE DB ' average score: ', '$'

scoreinfo DB ' highest score: ', '$',' lowest score: ', '$',' average score: ', '$'

DB ' 90-100: ', '$',' 80-89: ', '$',' 60-79: ', '$',' 0-60: ', '$',1 dup('|')

;用于数据存储

sname db 255, 0, 255 dup('$'); 学生姓名每人5个字符，每个字符以$结尾，最后加上|

sid db 255, 0, 255 dup('$'); 学号，5个字符，每个字符串以$结尾，最后加上|

scores db 255, 0, 255 dup('$');每次的成绩3个字符，每个字符串以$结尾,最后加上|

scoresresult db 255,0,255 dup('|')

final db 255, 0, 255 dup('$');加权总成绩的ascii,每个字符串以$结尾,最后加上\

finalresult db 255,0,255 dup('|');加权总成绩的十进制，中间无符号，只最后以\结尾

rankresult db 20,10,01,02,03,04,05,06,07,08,09,10,10 dup('|')

srankresult db 255,0,255 dup('|')

srank db 255,0,20 dup('|')

hla4result db 255,0,255 dup('|');hla:high low average 4 segment

hla4 db 255,0,255 dup('|');hla:high low average 4 segment

data ends

code segment

start:

mov dx,offset final

mov ax, data

mov ds,ax

system: ;绘制系统界面，进行交互

call anotherline

mov dx,offset T;ds:dx=需要输出字符串的首地址，字符串以’$’为结束标志。

mov ah,9;int 21h的9号功能调用（显示字符串）

int 21H

call anotherline

mov dx,offset LINE

mov ah,9

int 21H

call anotherline

mov dx,offset LABEL1

mov ah,9

int 21H

call anotherline

mov dx,offset LABEL2

mov ah,9

int 21H

call anotherline

mov dx,offset LABEL3

mov ah,9

int 21H

call anotherline

mov dx,offset LABEL4

mov ah,9

int 21H

call anotherline

mov dx,offset LABEL5

mov ah,9

int 21H

call anotherline

mov dx,offset LINE2

mov ah,9

int 21H

call anotherline

MOV AH, 01H

INT 21H

; 处理用户输入,转跳对于程序

CMP AL, '1'

JE toINPUT

CMP AL, '2'

JE toQUERY

CMP AL, '3'

JE toSORT

CMP AL, '4'

JE toSTAT

CMP AL, '5'

JE EXIT

JMP EXIT

toINPUT:

call anotherline

call far ptr INPUT

call anotherline

jmp system;执行完毕后又回到主界面

toQUERY:

call anotherline

call far ptr QUERY

call anotherline

jmp system;执行完毕后又回到主界面

toSORT:

call anotherline

call far ptr SORT

call anotherline

jmp system;执行完毕后又回到主界面

toSTAT:

call anotherline

call far ptr STAT

call anotherline

jmp system;执行完毕后又回到主界面

EXIT:

MOV AH, 4CH

INT 21H

;===================================

;=================函数===============

;===================================

anotherline:;换行

mov dl,0dh;回车键的ASCII码是0dh

mov ah,2;int 21h的2号功能调用（显示输出）

int 21h ;显示回车

mov dl,0ah

int 21h ;显示换行

ret

;=================信息录入+计算最终成绩程序===============

INPUT:

push ax

push bx

push cx

push dx

push si

push di

MOV AX, data ; 初始化数据段寄存器

MOV DS, AX

;bl储存着输入了几个学生，用于计算存放信息的位置

mov bx,0

mov cx,10;必须连续输入10个学生

input\_all\_stu:

inc bl

push bx;避免下面用到,先存好

push cx

call anotherline

input\_name:

mov dx,offset msg1;Please enter student name

mov ah,9h

int 21h

;mov dx,offset sname;这里不要加2，它会自己从第三字节开始写入的

mov dx,offset sname

mov al,bl

sub al,1

mov cl,6

mul cl

add dx,ax

mov ah,0ah;读字符串

int 21h

cmp bx,1

je input\_id;第1一个学生就不能改在前面加$了

mov si,dx

add si,1;本来应减1，但是上面是offset sname而非offset sname+2

mov byte ptr ds:[si], '$'; 不知道为什么前面总是有不对的符号结尾,所以这里是为了上个字符以$结尾

add si,6

mov byte ptr ds:[si], '$'

add si,1

mov byte ptr ds:[si], '|';最后一个名字后面加上｜，代表这是最后

input\_id:

call anotherline

mov dx,offset msg2;Please enter student id

mov ah,9h

int 21h

;mov dx,offset sid

mov dx,offset sid

mov al,bl

sub al,1

mov cl,6

mul cl

add dx,ax

mov ah,0ah;读字符串

int 21h

cmp bx,1

je input\_scores;第1一个学生就不能改在前面加$了

mov si,dx

add si,1;本来应减1，但是上面是offset sname而非offset sname+2

mov byte ptr ds:[si], '$'; 不知道为什么前面总是有不对的符号结尾,所以这里是为了上个字符以$结尾

add si,6

mov byte ptr ds:[si], '$'

add si,1

mov byte ptr ds:[si], '|';最后一个名字后面加上｜，代表这是最后

input\_scores:

call anotherline

mov dx,offset msg3;Please enter student scores

mov ah,9h

int 21h

mov si,offset scores+2;请注意这里的score只是作为缓冲区接收ascii，所以不用计算位置

mov cx, 16;16次作业

input\_scores1:

;由于使用mov ah, 0ah会重叠在同一行所以使用01h

mov ah, 01h

int 21h

mov ds:[si], al ; 将输入字符保存到数组中

inc si

mov ah, 01h

int 21h

mov ds:[si], al ; 将输入字符保存到数组中

inc si

mov ah, 01h

int 21h

mov ds:[si], al ; 将输入字符保存到数组中

inc si

mov byte ptr ds:[si], '$'

inc si

cmp cx,1

je input\_scores1end;为了最后一次没有逗号

mov ah, 02h;每次作业以逗号隔开

mov dl, ','

int 21h

loop input\_scores1

input\_scores1end:nop

call anotherline

mov dx,offset msg4;Please enter student final

mov ah,9h

int 21h

mov ah, 01h;继续用上面的ds:si

int 21h

mov ds:[si], al ; 将输入字符保存到数组中

inc si

mov ah, 01h

int 21h

mov ds:[si], al ; 将输入字符保存到数组中

inc si

mov ah, 01h

int 21h

mov ds:[si], al ; 将输入字符保存到数组中

inc si

mov byte ptr ds:[si], '$'

inc si

mov byte ptr ds:[si],'|'

inc si

call anotherline

;把scores内的ascii化成十进制，存入scoresresult

mov di,offset scoresresult+2

mov ax,0

mov al,bl

sub al,1

mov cl,17;每个学生17次成绩

mul cl

add di,ax;计算

mov si,offset scores+2;字符是从第3个字节开始的

; push di

; push si

call convertscoretonum

; pop di

; pop si;两个函数的di、si恰好相反

;call convertscoreToAscii;其实这个可有可无

;=================计算最终成绩主函数===============

;计算最终成绩

;循环16次，计算平时成绩，总平时成绩乘以4

;加上大作业成绩乘以6，再总体除以10

finalscore:

push ax

push bx

push cx

push dx

push si

push di

; 初始化寄存器

; mov si,offset scoresresult+2;每次的成绩

; mov di,offset finalresult+2;加权总成绩的十进制

mov si,offset scoresresult+2

mov ax,0

mov al,bl

sub al,1

mov cl,17

mul cl

add si,ax;计算si:+n学生\*17

mov di,offset finalresult+2

add di,bx;计算di:+(n-1学生)

sub di,1

push bx

onefinal:

mov ax,0

mov bx,0

mov cx,0

mov dx,0

mov cx,16;计算平时成绩，存储在bx中

normal:

mov ax,0

mov al,byte ptr ds:[si]

add bx,ax

inc si

loop normal

;平时成绩在bx

mov ax,bx;总平时成绩除以16

mov cl,16

div cl

mov bx,0

mov bl,al

mov ax,bx;平时成绩乘以4

mov cx,4

mul cx

mov bx,ax

mov ax,0;计算大作业成绩

mov al,byte ptr ds:[si];取出期末成绩

mov cx,6;期末成绩乘以6

mul cx

add bx,ax;加入bx

mov ax,bx;总成绩除以10

mov cx,10

div cx

mov byte ptr ds:[di],al

pop bx

;用余数把3位十进制数转化为ascii

mov si,offset finalresult+2

add si,bx;计算di:+(n-1学生)

sub si,1

mov di,offset final+2

mov ax,0

mov al,bl

sub al,1

mov cl,4

mul cl

add di,ax;计算si:+n学生\*4(每个学生4个字符)

; ToAscii:

; ToAsciibegin:

mov ax,0

mov al,byte ptr ds:[si];被除数放在ax

mov cl,100

div cl;除数为8位，al存储商，ah存储余数

add al,'0';商al即为百位的值，加上‘0’变成ascii码值

mov byte ptr ds:[di],al

inc di

mov al,ah;继续用余数ah

mov ah,0

mov cl,10

div cl

add al,'0';商al即为十位的值，加上‘0’变成ascii码值

mov byte ptr ds:[di],al

inc di

add ah,'0';余数ah即为个位的值，加上‘0’变成ascii码值

mov byte ptr ds:[di],ah

inc di

mov byte ptr ds:[di],'$';以保证每条字符串后面跟'$'

pop di

pop si

pop dx

pop cx

pop bx

pop ax

;=================计算最终成绩主函数结束===============

pop cx

pop bx

sub cx,1

cmp cx,0

jne toinput\_all\_stu

je endinput\_all\_stu

toinput\_all\_stu:

jmp far ptr input\_all\_stu

endinput\_all\_stu:nop

pop di

pop si

pop dx

pop cx

pop bx

pop ax

retf

;=================信息录入程序主函数结束===============

;将所有成绩的ascii转化为数字

; mov si,offset scores+2;字符是从第3个字节开始的

; mov di,offset scoresresult+2

convertscoretonum:

push ax

push bx

push cx

push dx

push si

push di

convertnumbegin:

mov dx,0

convertnum\_loop:

; 将当前字符转化为数字

mov bl,ds:[si] ; bl= 当前字符的 ASCII 码

sub bl,'0' ; ASCII码转换为数字

;如果当前不是最后一个字符，将面已经转化的数字乘以 10，并加上当前数字,存到dx

mov ax,0

mov al,dl

mov ah,10

mul ah

mov dx,ax

add dl,bl

; 检查还有没有下一位

inc si

cmp byte ptr ds:[si],'$'

je convertnum\_done

jmp convertnum\_loop;继续转化下一位

convertnum\_done:

mov ds:[di],dl;保存字符串

add di,1

add si,1

;一个字符串已经转化完成并保存，查看后面有没有字符串

cmp byte ptr ds:[si],'|'

jne convertnumbegin;非'|'说明后面还有字符要继续转化

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

;用余数把3位十进制数成绩转化为ascii

; mov si,offset finalresult+2

; mov di,offset final+2

; mov si,offset scoresresult+2

; mov al,bl

; sub al,1

; mov cl,17

; mul cl

; add si,ax

; mov di,offset scores+2

convertscoreToAscii:

push ax

push bx

push cx

push dx

push si

push di

mov cx,17

convertscoreToAsciibegin:

push cx

mov ax,0

mov al,byte ptr ds:[si]

mov cl,100

div cl

add al,'0';商al即为百位的值，加上‘0’变成ascii码值

mov byte ptr ds:[di],al

inc di

mov al,ah;继续用余数ah

mov ah,0

mov cl,10

div cl

add al,'0';商al即为十位的值，加上‘0’变成ascii码值

mov byte ptr ds:[di],al

inc di

add ah,'0';余数ah即为个位的值，加上‘0’变成ascii码值

mov byte ptr ds:[di],ah

inc di

mov byte ptr ds:[di],'$';以保证每条字符串后面跟'$'

inc di

; 检查还有没有下一个要转化的3位十进制数

inc si

pop cx

loop convertscoreToAsciibegin

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

;=================信息录入程序副函数结束===============

;=================成绩查询程序===============

QUERY:

push ax

push bx

push cx

push dx

push si

push di

mov ax,data

mov ds,ax

mov es,ax

Searchchose:

mov dx,offset amsg1;询问'Search by name or by student ID:'

mov ah,9

int 21h

mov ah,1

int 21h

mov bl,al;al内存储的是接收的字符

call anotherline

cmp bl,'1'

je byname

cmp bl,'2'

je byid

byname:

mov dx,offset amsg2;'Please input name(1-10characters):'

mov ah,9

int 21h

mov dx,offset buf;一定要注意！不要buf+2，它会自动存到第3字节的

mov ah,0ah

int 21h

mov ax,data

mov es,ax

mov bl,0;作为计数器，方便定位名字

mov di,offset buf+2

mov si,offset sname+2

findbyname:

mov di,offset buf+2

inc bl

mov cx,4;姓名字长为5

CLD ;增地址方向

repz cmpsb

jnz istherenext

jz printfind;若所有字符都相同

istherenext:

mov si,offset sname+2

mov al,bl

mov ah,6;含'$'是6个字符

mul ah

add si,ax

cmp byte ptr ds:[si],'|';说明后面没有名字了

je notfindname

jne findbyname

notfindname:

call anotherline

mov dx,offset amsg4;'There is no this student'

mov ah,9

int 21h

jmp printok

printfind:

call anotherline

call printinfo

jmp printok

byid:

mov dx,offset amsg3;'Please input student ID(1-10 characters):'

mov ah,9

int 21h

mov dx,offset buf;一定要注意！不要buf+2，它会自动存到第3字节的

mov ah,0ah

int 21h

mov ax,data

mov es,ax

mov bl,0;作为计数器，方便定位名字

mov di,offset buf+2

mov si,offset sid+2

findbyid:

mov di,offset buf+2

inc bl

mov cx,4;id字长为5

CLD ;增地址方向

repz cmpsb

jnz istherenext2

jz printfind2;若所有字符都相同

istherenext2:

mov si,offset sid+2

mov al,bl

mov ah,6;含'$'是6个字符

mul ah

add si,ax

cmp byte ptr ds:[si],'|';说明后面没有名字了

je notfindid

jne findbyid

notfindid:

call anotherline

mov dx,offset amsg4;'There is no this student'

mov ah,9

int 21h

jmp printok

printfind2:

call anotherline

call printinfo

jmp printok

printok:

pop di

pop si

pop dx

pop cx

pop bx

pop ax

retf

;=================成绩查询程序主函数结束===============

;输出学生信息，bl代表着第几个学生

printinfo:

mov ax,data

mov ds,ax

call anotherline

stuname:

mov dx,offset LABEL10;' 1. student name: '

mov ah,9

int 21h

mov dx,offset sname+2

mov al,bl

sub al,1

mov cl,6

mul cl

add dx,ax

mov ah,9h

int 21h

call anotherline

stuid:

mov dx,offset LABEL6;2. student ID

mov ah,9

int 21h

mov dx,offset sid+2

mov al,bl

sub al,1

mov cl,6

mul cl

add dx,ax

mov ah,9h

int 21h

call anotherline

;先把对应成绩的数字转为字符串

mov si,offset scoresresult+2

mov al,bl

sub al,1

mov cl,17

mul cl

add si,ax

mov di,offset scores+2

call convertscoreToAscii

stuscores:

mov dx,offset LABEL7;3. student scores(17 times)

mov ah,9

int 21h

call anotherline

mov dx,offset scores+2

; mov al,bl

; sub al,1

; mov cl,4\*17

; mul cl

; add dx,ax这里不需要乘，直接缓冲区输出就行

mov cx,17

printallscore1:

mov ah,9h

int 21h

add dx,4

push dx

mov dl,' '

mov ah,2;int 21h的2号功能调用（显示输出）

int 21h ;显示回车

pop dx

loop printallscore1

call anotherline

stufinal:

mov dx,offset LABEL8

mov ah,9

int 21h

mov dx,offset final+2

mov al,bl

sub al,1

mov cl,4

mul cl

add dx,ax

mov ah,9h

int 21h

call anotherline

sturank:

mov dx,offset LABEL9

mov ah,9

int 21h

mov dx,offset srank+2

mov al,bl

sub al,1

mov cl,3

mul cl

add dx,ax

mov ah,9h

int 21h

ret

;=================成绩排序程序===============

SORT:

mov ax,data

mov ds,ax

call studentrank

;在srank写上每个学生的排名

mov si,offset rankresult+2

mov di,offset srankresult+2

mov al,1

mov cx,10

sranking:

mov bx,0

mov bl,ds:[si]

sub bl,1

mov ds:[di+bx],al

inc al

inc si

loop sranking

;2位十进制数转化为ascii

mov di,offset srankresult+2

mov si,offset srank+2

ToAscii:

ToAsciibegin:

mov ax,0

mov al,byte ptr ds:[di]

mov cl,10

div cl

add al,'0';商al即为十位的值，加上‘0’变成ascii码值

mov byte ptr ds:[si],al

inc si

add ah,'0';余数ah即为个位的值，加上‘0’变成ascii码值

mov byte ptr ds:[si],ah

inc si

mov byte ptr ds:[si],'$';以保证每条字符串后面跟'$'

inc si

; 检查还有没有下一个要转化的3位十进制数

inc di

cmp byte ptr ds:[di],'|'

je ToAsciiend

jne ToAsciibegin

ToAsciiend:

mov byte ptr ds:[si],'|'

mov cx,10

mov si,offset rankresult+2

printrankinfo:

mov bx,0

mov bl,ds:[si]

call onestuscore

call printinfo2

inc si

loop printrankinfo

mov dx, offset buf;向缓冲区输入字符，以实现让功能表稍后出现，以展示整个排序

MOV AH, 01H

INT 21H

retf

;=================成绩排序程序主函数结束===============

studentrank:

push ax

push bx

push cx

push dx

push si

push di

mov si,offset finalresult+2

mov di,offset rankresult+2

mov cx,10

ranking:

mov bx,0;bx最大为8

mov dx,cx

sub dx,1

rank1:

mov ax,0

mov al,ds:[si+bx]

cmp al,ds:[si+bx+1]

jb next;小于则不交换，排序由大到小

mov ah,ds:[si+bx+1];交换成绩

mov ds:[si+bx+1],al;交换成绩

mov ds:[si+bx],ah;交换成绩

mov ax,0

mov al,ds:[di+bx]

mov ah,ds:[di+bx+1];交换序号

mov ds:[di+bx+1],al;交换序号

mov ds:[di+bx],ah;交换序号

next:

inc bx

sub dx,1

cmp dx,0

je rankend

jne rank1

rankend:

sub cx,1

cmp cx,1

jne ranking

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

; ;输出学生信息，bl代表着第几个学生

printinfo2:

push ax

push bx

push cx

push dx

push si

push di

sturank2:

mov dx,offset LABEL15

mov ah,9

int 21h

mov ax,0

mov dx,offset srank+2

mov al,bl

sub al,1

mov cl,3

mul cl

add dx,ax

mov ah,9h

int 21h

stuname2:

mov dx,offset LABEL11

mov ah,9

int 21h

mov dx,offset sname+2

mov al,bl

sub al,1

mov cl,6

mul cl

add dx,ax

mov ah,9h

int 21h

stuid2:

mov dx,offset LABEL12

mov ah,9

int 21h

mov dx,offset sid+2

mov al,bl

sub al,1

mov cl,6

mul cl

add dx,ax

mov ah,9h

int 21h

stufinal2:

mov dx,offset LABEL14

mov ah,9

int 21h

mov dx,offset final+2

mov al,bl

sub al,1

mov cl,4

mul cl

add dx,ax

mov ah,9h

int 21h

call anotherline

stuscores2:

mov dx,offset LABEL13

mov ah,9

int 21h

mov dx,offset scores+2

mov cx,17

printallscore2:

mov ah,9h

int 21h

add dx,4

push dx

mov dl,' '

mov ah,2;int 21h的2号功能调用（显示输出）

int 21h ;显示回车

pop dx

loop printallscore2

call anotherline

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

;读取一位学生的17次成绩，转换成ascii码，写入score

onestuscore:

push ax

push bx

push cx

push dx

push si

push di

mov di,offset scoresresult+2

mov al,bl

sub al,1

mov cl,17

mul cl

add di,ax;di

mov si,offset scores+2;si

mov cx,17

scoreAsciibegin: ;用余数把3位十进制数转化为ascii

push cx

mov ax,0

mov al,byte ptr ds:[di]

mov cl,100

div cl

add al,'0';商al即为百位的值，加上‘0’变成ascii码值

mov byte ptr ds:[si],al

inc si

mov al,ah;继续用余数ah

mov ah,0

mov cl,10

div cl

add al,'0';商al即为十位的值，加上‘0’变成ascii码值

mov byte ptr ds:[si],al

inc si

add ah,'0';余数ah即为个位的值，加上‘0’变成ascii码值

mov byte ptr ds:[si],ah

inc si

mov byte ptr ds:[si],'$';以保证每条字符串后面跟'$'

inc si

inc di

pop cx

loop scoreAsciibegin

scoreAsciiend:

mov byte ptr ds:[si],'|'

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

;=================成绩排序程序结束===============

;=================成绩统计程序===============

STAT:

mov ax,data

mov ds,ax

;得到最高分最低分

call HighandLow

call stuaverage

call statistics

mov di,offset hla4result+2

mov si,offset hla4+2

call ToAscii2

printhla:

mov si,offset scoreinfo

mov di,offset hla4+2

mov cx,7;有7个东西要打印

showinfoname:

mov dx,si

mov ah,9h

int 21h

nextinfoname:

inc si

cmp byte ptr ds:[si],'$'

je showhla

jmp nextinfoname

showhla:

inc si;这样下次又可以打印score info

mov dx,di

mov ah,9h

int 21h

add di,4

call anotherline

loop showinfoname

retf

;=================成绩统计程序主函数结束===============

HighandLow:

push ax

push bx

push cx

push dx

push si

push di

mov si,offset finalresult+2

mov di,offset hla4result+2

mov cx,9;10个数只需比9次

mov bx,1

mov ax,0

mov al,ds:[si]

rankhigh:

cmp al,ds:[si+bx]

ja next1;al大则不交换，保留大的数在al

mov al,ds:[si+bx]

next1:inc bx

loop rankhigh

mov ds:[di],al;存入hla

inc di

mov cx,9

mov bx,1

mov ax,0

mov al,ds:[si]

ranklow:

cmp al,ds:[si+bx]

jb next2;al小则不交换，保留小的数在al

mov al,ds:[si+bx]

next2:inc bx

loop ranklow

mov ds:[di],al;存入hla

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

stuaverage:

push ax

push bx

push cx

push dx

push si

push di

mov ax,data

mov ds,ax

; 初始化寄存器

mov si,offset finalresult+2;所有学生的最终成绩

mov di,offset hla4result+4;平均成绩

mov cx,10;平均成绩总和存储在bx中

average1:

mov ax,0

mov al,ds:[si]

add bx,ax

inc si

loop average1

;平时成绩在bx

mov ax,bx;总平时成绩除以10

mov cl,10

div cl

;al为商，是平均成绩

mov ds:[di],al

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

statistics:

push ax

push bx

push cx

push dx

push si

push di

mov ax,data

mov ds,ax

; 初始化寄存器

mov si,offset finalresult+2;所有学生的最终成绩

mov di,offset hla4result+5;保存4个分数段的人数

mov byte ptr ds:[di],0

inc di

mov byte ptr ds:[di],0

inc di

mov byte ptr ds:[di],0

inc di

mov byte ptr ds:[di],0

mov di,offset hla4result+5

mov cx,10

statistics1:

mov ax,0

mov al,ds:[si]

cmp al,90;100-90

jb nextseg1;如果小于，比较下一个分数段

add byte ptr ds:[di],1

jmp statisticsend;如果大于，加1后结束该成绩的比较

nextseg1:

mov al,ds:[si]

cmp al,80;89-80

jb nextseg2

add byte ptr ds:[di+1],1

jmp statisticsend

nextseg2:

mov al,ds:[si]

cmp al,60;79-60

jb nextseg3

add byte ptr ds:[di+2],1

jmp statisticsend

nextseg3:

add byte ptr ds:[di+3],1

statisticsend:

inc si

loop statistics1

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

;用余数把3位十进制数转化为ascii

; mov di,offset finalresult+2

; mov si,offset final+2

ToAscii2:

push ax

push bx

push cx

push dx

push si

push di

ToAsciibegin2:

mov ax,0

mov al,byte ptr ds:[di]

mov cl,100

div cl

add al,'0';商al即为百位的值，加上‘0’变成ascii码值

mov byte ptr ds:[si],al

inc si

mov al,ah;继续用余数ah

mov ah,0

mov cl,10

div cl

add al,'0';商al即为十位的值，加上‘0’变成ascii码值

mov byte ptr ds:[si],al

inc si

add ah,'0';余数ah即为个位的值，加上‘0’变成ascii码值

mov byte ptr ds:[si],ah

inc si

mov byte ptr ds:[si],'$';以保证每条字符串后面跟'$'

inc si

; 检查还有没有下一个要转化的3位十进制数

inc di

cmp byte ptr ds:[di],'|'

je ToAsciiend2

jne ToAsciibegin2

ToAsciiend2:

mov byte ptr ds:[si],'|'

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

;=================成绩统计程序结束===============

code ends

end start

签名：

日期： 2023-6-23