实验名称：循环程序设计

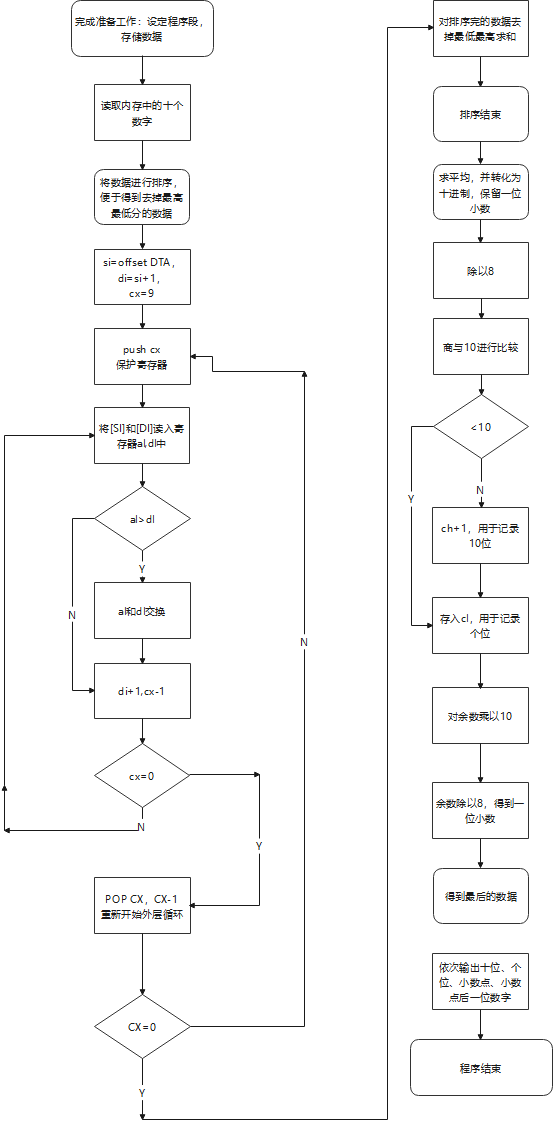
1. 实验目的
2. 熟练掌握8086/8088常用汇编指令的使用方法
3. 熟练掌握循环结构程序编程技巧
4. 熟练掌握汇编语言程序运行调试方法
5. 实验要求

认真分析实验题目，设计程序流程图，独立完成代码编写及运行调试

1. 实验题目

有10个评委评分，打分满分是100，去掉一个最高分，去掉一个最低分，求平均值，以十进制方式显示，保留小数点后一位数。

1. 程序流程图



1. 程序代码

dseg segment

DTA db 10,20,30,40,50,60,70,80,91,95

dseg ends

cseg segment

assume cs:cseg,ds:dseg;

start: mov ax,dseg;

mov ds,ax;

mov si,offset DTA;

mov cx,9;设定初始值

call Outorder

mov si,offset DTA

add si,1

mov ax,0

mov cx,8

;求和并转化为十进制

call SUM

;保护al寄存器

mov dh,al

;显示十位

mov dl,ch

add dl,30h

mov ah,02h

int 21h

;显示个位

mov dl,cl

add dl,30h

mov ah,02h

int 21h

;显示小数点

mov dl,2eh

mov ah,02h

int 21h

;显示小数点后一位的数字

mov dl,dh

add dl,30h

mov ah,02h

int 21h

;返回程序

mov ah,4ch

int 21h

;程序终止

;外排序程序

Outorder PROC

mov di,si;

inc di;di=si+1

mov bx,cx;保存外排序计数器

call Innerorder;调用内排序

back:

mov cx,bx;恢复外排序计数器

inc si;外排序指针加一

loop Outorder;循环直到cx为0

endOutorder:ret;排序结束返回主函数

Outorder ENDP

;内排序程序

Innerorder PROC

mov al,[si]

mov dl,[di]

cmp al,dl

;如果A<B

jnae next

;交换

xchg al,dl

next:

mov [si],al

mov [di],dl

inc di;di+1

loop Innerorder

jmp back;返回外排序

Innerorder ENDP

;对总成绩求和并换算为十进制

SUM PROC

add al,[si];

adc ah,0;

inc si;

loop SUM

mov bl,8

div bl;商存在al,余数在ah

xchg al,ah;余数和商位置调换，商存在ah,余数在al

;初化循环

;ch置零,最后用于存十位

mov ch,0

Todec:

cmp ah,10

js next2

sub ah,10

;逢十进一

inc ch

jmp Todec

next2:

;个位放在cl里

mov cl,ah

;将余数转化为一位小数,余数存在ah里,这里可以知道余数必定小于08h，所以只需对余数乘以10，除以8，取其第一位即可

mov bl,10

mul bl

;ax为乘以10之后的余数，将它除以8，商存在al里，也是十进制显示

div bl

endSUM:ret;返回主函数

SUM ENDP

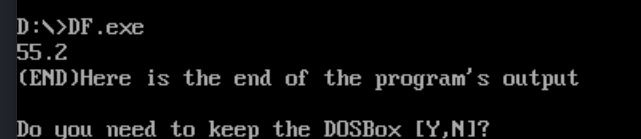
cseg ends

end start

1. 实验总结

实验中遇到的问题和解决方法：

1. 数据的存储问题：要求存十个数字，范围是0-100，存储时，如果采用16进制的00h-100h存储，那么会有一些问题，如100h的问题的表示问题以及80h以上的数字就是有符号数了，如果后续比较大小会被编译器识别为负数，那么比较大小就不太容易，解决方案：采用十进制存储数据，虽然存入的还是十六进制，但是转化十进制可以在后期实现
2. 求最大值和最小值，这里我虽然使用了冒泡排序的方式，然后再从1到9读取数据，但是后来想到这样算法的复杂程度过高，为O（n2）。考虑到只要得到最大值和最小值，所以一个循环就可以解决，每次循环中记录当前的最大值和最小值即可，算法复杂度为O（n）,然后对十个数的和再减最大值和最小值即可。
3. 在子程序中调用子程序出现bug，由于这个实验是一个相对实际的实验，所以多个子程序是必要的，这样可以提升程序的健壮性，对于汇编语言的子程序而言，要确定好他们编写的位置以及执行完后要返回的位置，解决方式：将子程序写在主程序后边，这样主程序调用程序结束之后，子程序不会再执行一遍，对于子程序中调用子程序的bug，采用jmp的形式调回上一层即可，而不能采用ret直接回到主程序。
4. 将最后的结果转化成十进制出现问题，解决方案：这里考虑到先除以8再转化成十进制还是先转化，再除以8，由于数字过大（需要用AX来存储），如果先转化则特别麻烦，可以发现，除以8之后的商可以仅用al来存储，这样的话就先进行除以8的操作，再捡出商和余数即可。
5. 如何转化为十进制的问题：正常情况下采用十进制调整指令DAA即可，但注意变量存储时采用的是十进制变化为十六进制以后的变量存储，所以必须采用其他的方法，解决方案：采用最基本的“逢十进一”的思想，对商进行减0ah的操作，每有一个0ah，就记录十位加一，最后不够减的就全是个位了。
6. 如何保留小数点后一位：解决方案;采用笔算除法的思想，对于余数，乘以10，再进行除以8的操作，得到的商就是第一位小数，得到的余数甚至可以进一步进行这种操作来得到第二位、第三位小数，完成四舍五入等精细的操作。
7. 显示问题，将数据更为精细的显示，解决方案是将数据分别存入各个寄存器中，然后+30h显示，中间再显示一个小数点，采用ASCII码实现



8.显示字符时，出现有字符显示错误的问题，解决方式:显示错误说明存在寄存器中的值错误了，但并没有对寄存器进行过错误的操作，经过研究后发现，采用02h显示之后，不仅是dl，al的值也会受到影响，而存在al寄存器中的值就会出现问题，所以我才用堆栈来保护al。