

# 汇编作业报告

## 题目一

### 原题

11个同学的汇编成绩分别为（括号中为学号）：

89 (01) ,76 (02) ,91 (03) ,93 (04) ,58 (05) ,64 (06) ,83 (07) ,81 (08) ,45 (09) ,72 (10) ,84 (11)

- (1) 统计成绩位于100~90,89~80,79~70,69~60,59~0等区间的学生人数
- (2) 输出最高分，最低分，以及11个同学的平均成绩
- (3) 附加要求：将11个学生分数从高到低排序并将排序后的成绩和对应的学号输出

### 源代码

```
DATA SEGMENT
    SCORE DB 89,76,4,93,58,64,83,81,45,72,84
    NUM    DB 01,02,03,04,05,06,07,08,09,10,11
    SUM    DW 0
    AVE     DB 0
    CRLF   DB 0AH,0DH,'$'
    SC     DB '100~90',09H,'89~80',09H,'79~70',09H,'69~60',09H,'59~0',0DH,0AH,'$'
    MAXS   DB 0AH,0DH,'MAX=','$'
    MINS   DB 0AH,0DH,'MIN=','$'
    AVES   DB 0AH,0DH,'AVE=','$'
DATA ENDS

STK SEGMENT;堆栈段
    DB 20 DUP(?)
STK ENDS

CODE SEGMENT
    ASSUME DS:DATA,CS:CODE,SS:STK

MAIN PROC FAR

START:
    ;初始化
    MOV AX,DATA
    MOV DS,AX
    MOV AX,STK
    MOV SS,AX

    CALL SORT
    CALL PRINTTAB
```

```

;输出最高分
LEA DX,MAXS
MOV AH,09H
INT 21H

XOR AX,AX
XOR BX,BX
LEA SI,SCORE
MOV AL,[SI]
CALL PRINT

;输出最低分
LEA DX,MINS
MOV AH,09H
INT 21H

XOR AX,AX
XOR BX,BX
LEA SI,SCORE
MOV AL,[SI+10]
CALL PRINT

;输出平均成绩
LEA DX,AVES
MOV AH,09H
INT 21H
;求和
XOR AX,AX
XOR BX,BX
XOR DX,DX
LEA SI,SCORE
MOV CX,11
LP:
MOV BL,[SI]
ADD AX,BX
INC SI
LOOP LP
LEA DI,SUM
MOV [DI],AX;和存储到SUM中

MOV CX,0BH
DIV CX
LEA DI,AVE
MOV [DI],AX

CALL PRINT

LEA DX,CRLF;换行
MOV AH,09H
INT 21H

;输出成绩排序结果
LEA DI,SCORE
MOV SI,DI
MOV CX,11
LP1:
XOR AX,AX

```

```

XOR BX,BX
XOR DX,DX
MOV AL,[SI]
CALL PRINT

MOV DL,20H;空格
MOV AH,2
INT 21H

INC SI
LOOP LP1
;输出成绩排序结果结束

LEA DX,CRLF;换行
MOV AH,9
INT 21H

MOV CX,11
LEA SI,NUM

```

LP2:

```

XOR AX,AX
MOV AL,[SI]
CALL PRINT

MOV DL,20H;空格
MOV AH,2
INT 21H

INC SI
LOOP LP2
;输出成绩排序后对应的学号

MOV AH,4CH
INT 21H

```

MAIN ENDP

SORT PROC NEAR

```

MOV BL,10;BL为外循环计数器
LEA DI,SCORE;DI存放成绩初始地址

```

;外循环

NEXT1:

```

MOV SI,DI
MOV CL,BL;CL为内循环计数器

```

;内循环

NEXT2:

```

MOV AL,[SI]
INC SI
CMP AL,[SI]
JNC NEXT3;如果前>后跳过交换
;交换成绩
MOV AH,[SI]
MOV [SI-1],AH
MOV [SI],AL
;交换编号

```

```
MOV AX,SI
SUB AX,DI
ADD AX,OFFSET NUM
PUSH SI;保护SI数据
MOV SI,AX
MOV AH,[SI]
MOV AL,[SI-1]
MOV [SI],AL
MOV [SI-1],AH
```

```
POP SI;返回SI之前的位置
```

```
NEXT3:
DEC CL
JNZ NEXT2
;内循环结束
```

```
DEC BL
JNZ NEXT1
;外循环结束
RET
```

```
SORT ENDP
```

```
PRINTTAB PROC NEAR
```

```
;输出区间人数
LEA DX,SC;输出分数段字符串
MOV AH,09H
INT 21H

LEA DI,SCORE
MOV SI,DI
MOV CX,4;循环4次
XOR AX,AX
XOR DX,DX
MOV DH,90;DH存储区间下界
```

```
CP:
CMP [SI],DH
JC PUT
INC SI;指针右移
INC DL;DL存储区间人数
JMP CP
```

```
PUT:;小于DH 输出人数
```

```
XOR AX,AX
MOV AL,DL
MOV DL,0AH
DIV DL
MOV BX,AX
MOV DL,BL
ADD DL,30H
MOV AH,02H
INT 21H
MOV DL,BH
ADD DL,30H
INT 21H
```

```
MOV DL,09H;制表符号
```

```
MOV AH,02H
```

```
INT 21H
```

```
XOR DL,DL
```

```
SUB DH,10
```

```
LOOP CP
```

```
MOV AX,SI
```

```
SUB AX,OFFSET SCORE
```

```
MOV BX,11
```

```
SUB BX,AX
```

```
MOV AL,BL
```

```
MOV DL,0AH
```

```
DIV DL
```

```
MOV BX,AX
```

```
MOV DL,BL
```

```
ADD DL,30H
```

```
MOV AH,02H
```

```
INT 21H
```

```
MOV DL,BH
```

```
ADD DL,30H
```

```
INT 21H
```

```
RET
```

```
PRINTTAB ENDP
```

```
PRINT PROC NEAR
```

```
MOV DL,0AH
```

```
DIV DL
```

```
MOV BX,AX
```

```
MOV DL,BL
```

```
ADD DL,30H
```

```
MOV AH,02H
```

```
INT 21H
```

```
MOV DL,BH
```

```
ADD DL,30H
```

```
INT 21H
```

```
RET
```

```
PRINT ENDP
```

```
CODE ENDS
```

```
END START
```

## 运行结果

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file: 1.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 463k

D:\>if exist 1.OBJ tlink /v/3 1.obj >X:\LINK.LOG

Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
Warning: No stack

D:\>1.exe
100~90 89~80 79~70 69~60 59~0
01 04 02 01 03
MAX=93
MIN=04
04 01 11 07 08 02 10 06 05 09 03
(END)Here is the end of the program's output

Do you need to keep the DOSBox [Y,N]?_
```

## 算法分析

1. SORT 函数实现排序工作
2. PRINTTAB 函数实现对成绩表的打印
3. PRINT 函数实现对某个成绩的打印

## 题目二

### 原题

对随机输入的 11 名学生的成绩进行排序与分数段人数统计，输出最高分，最低分，中间值以及排序后的成绩。

内容：定义存储空间存放成绩，输出不及格人数，60~69 分数段人数，.....， 90~100 分数段人数，对成绩进行排序，输出排序后的

结果，在显示器上显示如下信息：

The score between 90 and 100 : xx

.....

The score between 0 and 59 : xx

Min is xx

Max is xx

The middle score is xx

Rank ordering : xx xx xx,.....

xx表示一个十进制数

## 源代码

```
DATA SEGMENT
    SCORE DB 11 DUP(?)
    S      DB 'The score between ','$'
    S1     DB 'The score between 0 and 59 : ','$'
    AD     DB ' and ','$'
    CO     DB ' : ','$'
    M1     DB 'Min is ','$'
    M2     DB 'Max is ','$'
    M3     DB 'Mid is ','$'
    WT     DB 'Please input 11 scores separated by space: ',0AH,0DH,'$'
    RANK   DB 'Rank order : ','$'
    CRLF   DB 0AH,0DH,'$'
DATA ENDS

STK SEGMENT
    DB 20 DUP(?)
STK ENDS

CODE SEGMENT
    ASSUME DS:DATA,SS:STK,CS:CODE

MAIN PROC FAR

START:
    ;初始化
    MOV AX,DATA
    MOV DS,AX
    MOV AX,STK
    MOV SS,AX

    ;输入11名学生成绩
    LEA DX,WT
    MOV AH,09H
    INT 21H

    MOV CX,11;输入11个字符
LP: MOV AH,1;1号功能 输入一个字符
    INT 21H
    CMP AL,' ';和空格比较
    JZ LP;如果空格重新输入

    ;分析两个空格中的数字大小
    SUB AL,30H
    MOV DH,AL;把第一位数放到DH中
    MOV BL,0AH

CUL:
    MOV AH,1
    INT 21H
    CMP AL,' '
```

```
JZ PUT
SUB AL,30H
MOV DL,AL
MOV AH,0
MOV AL,DH
MUL BL
ADD AL,DL
MOV DH,AL
JMP CUL
```

PUT:

```
MOV [SI],DH
INC SI
LOOP LP
;输入成绩结束
```

```
LEA DX,CRLF
MOV AH,09H
INT 21H
```

```
CALL SORT
```

;输出区间人数

;初始化

```
LEA DI,SCORE;DI存储分数段首地址
```

```
MOV SI,DI
```

```
XOR AX,AX
```

```
XOR BX,BX
```

```
MOV BL,90;区间下界
```

```
MOV CX,4;先输出前四个区间人数
```

```
XOR DX,DX
```

JUD:

```
PUSH CX;保护CX之前的数据
```

```
CMP [SI],BL
```

```
JC SC;如果小于当前区间下界跳转到SC输出区间人数
```

```
INC BH;BH存储当前区间人数
```

```
ADD SI,1
```

```
JMP JUD
```

;输出当前区间人数

SC: LEA DX,S

```
MOV AH,09H
```

```
INT 21H
```

;输出区间下界

```
XOR CX,CX
```

```
MOV AL,BL
```

```
CALL DCM;调用
```

```
LEA DX,AD
```

```
MOV AH,09H
```

```
INT 21H
```

;输出区间上界

```
XOR CX,CX
```

```
MOV AL,BL
```

```
ADD AL,9
```

```
CMP AL,99
```

```
JNZ OT;不是99直接输出
```



ADD AL,1;如果是99那就再加一

OT: CALL DCM

LEA DX,CO  
MOV AH,09H  
INT 21H

XOR CX,CX  
MOV AL,BH  
CALL DCM

LEA DX,CRLF  
MOV AH,09H  
INT 21H

SUB BL,10;区间下界减10  
XOR BH,BH  
POP CX  
DEC CX  
JNZ JUD;返回继续判断之后的区间数据个数

;输出0~59区间  
LEA DX,S1  
MOV AH,09H  
INT 21H

MOV BX,SI  
SUB BX,OFFSET SCORE;大于59的数的个数  
MOV AX,11  
SUB AX,BX;0~59区间的个数  
CALL DCM;以十进制输出

;输出最大、最小、中间值

LEA DX,CRLF;换行  
MOV AH,09H  
INT 21H

;输出最小值

LEA DX,M1  
MOV AH,09H  
INT 21H

LEA SI,SCORE  
MOV AL,[SI+10]  
CALL DCM

LEA DX,CRLF;换行  
MOV AH,09H  
INT 21H

;输出最大值

LEA DX,M2  
MOV AH,09H  
INT 21H

LEA SI,SCORE  
MOV AL,[SI]  
CALL DCM

```

    LEA DX,CRLF;换行
    MOV AH,09H
    INT 21H

;输出中间值
    LEA DX,M3
    MOV AH,09H
    INT 21H

    LEA SI,SCORE
    MOV AL,[SI+5]
    CALL DCM

    LEA DX,CRLF;换行
    MOV AH,09H
    INT 21H

;输出成绩排序结果
    LEA DX,RANK;提示语
    MOV AH,09H
    INT 21H

    LEA DI,SCORE;把成绩偏移地址给DI
    MOV SI,DI;DI把成绩偏移地址给SI
    MOV CX,11
LPP:
    PUSH CX
    XOR CX,CX
    XOR DX,DX
    MOV AL,[SI]
;以十进制输出
    CALL DCM

    MOV DL,20H;空格
    MOV AH,2
    INT 21H

    INC SI
    POP CX
    LOOP LPP
;输出成绩排序结果结束

    MOV AH,4CH;程序结束
    INT 21H
MAIN ENDP

SORT PROC NEAR

;开始排序
    MOV BL,10;BL为外循环计数器
    LEA DI,SCORE;DI存放成绩初始地址

;外循环
NEXT1:
    MOV SI,DI;把成绩初始地址给SI
    MOV CL,BL;CL为内循环计数器

```

```

        ;内循环
NEXT2:
    MOV AL,[SI];把SI指向的成绩赋值给AL
    INC SI;指针右移
    CMP AL,[SI];后一个成绩和前一个成绩比较
    JNC NEXT3;如果前>后跳过交换
    ;交换成绩
    MOV AH,[SI]
    MOV [SI-1],AH
    MOV [SI],AL

NEXT3:
    DEC CL
    JNZ NEXT2
    ;内循环结束

    DEC BL
    JNZ NEXT1
    ;外循环结束
    RET

SORT ENDP

DCM PROC NEAR

    PUSH CX
T1: MOV AH,0
    MOV DL,0AH
    DIV DL
    PUSH AX
    INC CX
    CMP AL,0
    JNZ T1

T2: POP DX
    MOV DL,DH
    ADD DL,30H
    MOV AH,02H
    INT 21H
    LOOP T2

    POP CX
    RET
DCM ENDP

CODE ENDS
    END START

```

## 运行结果

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Passes: 1
Remaining memory: 462k

D:\>if exist 2.OBJ tlink /v/3 2.obj >X:\LINK.LOG

Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
Warning: No stack

D:\>2.exe
Please input 11 scores separated by Space:
1 2 3 4 5 6 7 8 99 100 11
The score between 90 and 100 :2
The score between 80 and 89 :0
The score between 70 and 79 :0
The score between 60 and 69 :0
The score between 0 and 59 :9
Min is 1
Max is 100
The middle score is 6
Rank ordering : 100 99 11 8 7 6 5 4 3 2 1
(END)Here is the end of the program's output

Do you need to keep the DOSBox [Y,N]?
```

## 算法分析

1. 延续使用作业一中的SORT函数，进行排序
2. 增加DCM函数对输入数值进行转换

## 题目三

### 原题

- 1、输入两个十进制的数，以十六进制输出两者的和，例如，输入3和4，输出07；输入6和7，输出0D；输入8和9，输出11。
- 2、输入两个十六进制的数，以十进制输出两者的和。例如，输入3和5，输出08；输入6和7，输出13；输入9和A，输出19。

### 源代码

```
DATA SEGMENT
    NUM1 DB 0,0
    NUM2 DB 0,0
    SUM DW 0,0
    STR1 DB 'Please input two numbers in DEC:',0AH,0DH,'$'
    SUM1 DB 'Sum in HEX is:',0AH,0DH,'$'
    STR2 DB 'Please input two numbers In HEX:',0AH,0DH,'$'
    SUM2 DB 'Sum in DEC is:',0AH,0DH,'$'
    CRLF DB 0AH,0DH,'$'
DATA ENDS
```

```

STK SEGMENT
    DB 128 DUP(0)
STK ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:STK
START:
    MOV AX,DATA
    MOV DS,AX
    MOV AX,STK
    MOV SS,AX
    LEA DX,STR1
    MOV AH,09
    INT 21H

IN1:
    MOV AH,01H
    INT 21H
    MOV NUM1[0],AL
    SUB NUM1[0],30H

    MOV AH,01H
    INT 21H
    MOV NUM1[1],AL
    SUB NUM1[1],30H

    LEA DX,CRLF
    MOV AH,9
    INT 21H

AD1:
    MOV BL,NUM1[0]
    ADD BL,NUM1[1]
    MOV SUM[0],BX

RESHEX:
    LEA DX,SUM1
    MOV AH,9
    INT 21H

    MOV AX,SUM[0]
    CMP AX,16
    JG OUTNUM
    CMP AX,10
    JB OUTNUM
    JMP OUTCHAR

OUTNUM:
    MOV BX,16
    MOV CX,0
    MOV DX,0

S1:
    DIV BX
    PUSH DX
    INC CX
    CWD
    CMP AX,0
    JNE S1

```

S2:

```
POP DX
ADD DL,30H
MOV AH,02H
INT 21H
LOOP S2
JMP OUTHEX
```

OUTCHAR:

```
MOV DX,0
SUB AX,10
MOV DL,AL
ADD DL,65
MOV AH,02H
INT 21H
JMP OUTHEX
```

OUTHEX:

```
LEA DX,CRLF
MOV AH,09
INT 21H
LEA DX,STR2
MOV AH,9
INT 21H
MOV AH,01H
INT 21H
MOV NUM2[0],AL
MOV AH,01H
INT 21H
MOV NUM2[1],AL

LEA DX,CRLF
MOV AH,9
INT 21H
LEA DX,SUM2
MOV AH,9
INT 21H
```

JUD PROC NEAR

```
MOV CX,2
MOV SI,0
MOV BX,0
```

ADD2:

```
MOV AL,NUM2[SI]
CMP AL,57
JG TO_DEC
```

ADD\_AGAIN:

```
SUB NUM2[SI],30H
```

AD: INC SI

```
LOOP ADD2
JMP RESULT_DEC
```

TO\_DEC:

```
SUB AL,65
ADD AL,48
```

```
    SUB AL,38
    SUB AL,32
    MOV NUM2[SI],AL
    JMP AD
    RET
JUD ENDP
```

RESULT\_DEC:

```
    MOV BL,NUM2[0]
    ADD BL,NUM2[1]
    MOV SUM[2],BX
    JMP OUTNUM2
```

OUTNUM2 PROC NEAR

```
    MOV AX,BX
    MOV BX,10
    MOV CX,0
    MOV DX,0
```

T1:

```
    DIV BX
    PUSH DX
    INC CX
    CWD
    CMP AX,0
    JNE T1
```

T2:

```
    POP DX
    ADD DL,30H
    MOV AH,02H
    INT 21H
    LOOP T2
    JMP EXIT
    RET
```

OUTNUM2 ENDP

EXIT:

```
    MOV AX,4C00H
    INT 21H
```

```
CODE ENDS
END START
```

## 运行结果

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Assembling file: 3.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 463k

D:\>if exist 3.OBJ tlink /v/3 3.obj >X:\LINK.LOG

Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
Warning: No stack

D:\>3.exe
Please Input Two Numbers In DEC:
12
Sum in HEX is:
3
Please Input Two Numbers In HEX:
aa
Sum in DEC is:
20
(END)Here is the end of the program's output

Do you need to keep the DOSBox [Y,N]?
```

## 算法分析

1. 程序分为两个部分，实现两种数制的相加和转换
2. 在十六进制时候，数字（0-9）和字母（a-f）时，都需要分别处理
3. 对于读入的字符，先用JUD函数判断为数字还是字母

## 题目四

### 原题

使用串操作指令 MOVSB 对一段内存单元中的内容（1，2，3，.....，100）进行转移，再使用串操作指令 CMPS 对转移的内容进行

比较来判断传输是否正确，若不正确则进行重新传输；接着对已经正确传输的 100 个数据进行无符号型的累加，最后使用 BCD 调整

码，最终将答案放入内存，并将其显示在屏幕上。

### 源代码

```
DATA SEGMENT;定义数据段
    MKEY DB 0DH,0AH,'MATCH',0DH,0AH,'$';定义匹配时的输出字符
    NMKEY DB 0DH,0AH,'NOMATCH',0DH,0AH,'$';定义不匹配时的输出字符
    SUM DW ?
DATA ENDS;数据段结束
CODE SEGMENT;定义代码段
    ASSUME CS:CODE;说明代码段和数据段的位置
```



START:

;初始化

```
MOV AX,1000H
MOV DS,AX;定义数据段地址
MOV AX,1
MOV CX,100;初始化AX和CX
S:
MOV [DI],AX
INC AX
INC DI
LOOP S;将1~100循环存入DS
```

;转存部分

```
MOV DI,0;清空目的指针
MOV AX,2000H
MOV ES,AX;定义附加段地址
MOV CX,100;设置循环次数
CLD;清空方向标志,从低地址向高地址移动
REP MOVSB;当CX不为0时继续配对
```

;比较部分

```
MOV CX,100;设置循环次数
REPZ CMPSB;当CX不为0且字符串匹配时继续配对
JNZ MATCH;ZF=1说明匹配成功
JZ NOMATCH;ZF=0说明匹配失败
```

;输出标志信息部分

MATCH:

```
MOV AX,DATA
MOV DS,AX;重新定义数据段
LEA DX,MKEY;取出MATCH的偏移地址
MOV AH,09H;调用09H指令输出MATCH
INT 21H
JMP A;跳转到累加部分
```

NOMATCH:

```
MOV AX,DATA
MOV DS,AX;重新定义数据段
LEA DX,NMKEY;取出NOMATCH的偏移地址
MOV AH,09H;调用09H指令输出NOMATCH
INT 21H
JMP A;跳转到累加部分
```

;累加部分

```
A:  MOV AX,1000H
MOV DS,AX;重新定义数据段
MOV CX,100
XOR AX,AX;清空AX
XOR SI,SI;清空SI
XOR DI,DI;清空DI
```

NEXT:

```
XOR AX,AX;清空AX
ADD AL,[SI];利用AL存放一个字节中的数据
MOV AH,0;修正高地址
INC SI;指针位移
ADD DX,AX;保证字长匹配,相加
LOOP NEXT;循环相加
```

;输出部分

```
MOV BX,DX;将结果转存入BX
MOV SI,0AH
XOR CX,CX;清空计数器
```

```

MOV AX,BX;将结果转存入AX,准备除法

L:   XOR DX,DX
     DIV SI;取出AX的末位数
     PUSH DX;余数入栈保护数据
     INC CX;计数加一,为输出做准备
     CMP AX,0
     JNZ L;直到商为0时停止循环

PRINT:POP DX;数据出栈,FILO顺序正确
      ADD DL,30h;转化成可以输出的ASCII
      MOV AH,2
      INT 21h;调用输出字符的02H指令
      LOOP PRINT

      MOV AX,4C00H
      INT 21H;返回程序

CODE ENDS;代码段结束

END START;结束程序

```

## 运行结果

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

D:\>tasm /zi 4.ASM >X:\ASM.LOG

Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file: 4.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 465k

D:\>if exist 4.OBJ tlink /v/3 4.obj >X:\LINK.LOG

Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
Warning: No stack

D:\>4.exe

MATCH
5050
(END)Here is the end of the program's output

Do you need to keep the DOSBox [Y,N]?_

```

## 算法分析

1. 利用循环取出每一位数字，调整为 ASCII，并用 02H 号指令输出 字符，通过循环输出每一个字符
2. 主要使用基本的串操作指令完成相关功能

# 题目五

## 原题

计算  $3X-2|Y|+|Z|$ ，并将计算结果输出显示。

## 源代码

```
DATA SEGMENT
    STRING1 DB "Enter 3 numbers",0AH,'$'
    STRING2 DB "Please input X:",'$'
    STRING3 DB "Please input Y:",'$'
    STRING4 DB "Please input Z:",'$'
    STRING5 DB "3X-2|Y|+|Z|= ",'$'
    X        DB 8 DUP(0)
    Y        DB 8 DUP(0)
    Z        DB 8 DUP(0)
    RESULT   DB 3 DUP(0)
    FLAG     DB 0
DATA ENDS

STK SEGMENT
    DB 16 DUP(0)
STK ENDS

CODE SEGMENT
    ASSUME CS:CODE,DS:DATA,SS:STK
MAIN PROC FAR
START:
    MOV AX,DATA
    MOV DS,AX
    MOV AX,STK
    MOV SS,AX
    MOV SP,16

    LEA BX,STRING5
    PUSH BX
    LEA BX,STRING4
    PUSH BX
    LEA BX,STRING3
    PUSH BX
    LEA BX,STRING2
    PUSH BX
    LEA BX,STRING1
    PUSH BX

    MOV SI,0
    MOV BP,10

    LEA BX,X
    POP DX      ;获取STRING1偏移地址
    MOV AH,09H
    INT 21H     ;显示提示语
```

```
MOV CX,3
```

INPUT:

```
POP DX
MOV AH,09H
INT 21H      ;每次显示一个输入提示，循环3次
CALL INCHAR   ;读取字符
MOV AL,[BX]
MOV AH,0      ;AL存放首位字符，开始准备计算数值
PUSH BX
PUSH CX
SUB SI,2
JZ SOLO      ;个位数直接算
MOV CX,SI
MOV SI,1
CALL EXC     ;字符转化为数AL
```

S:

```
POP CX
POP BX
MOV BYTE PTR[BX],AL
ADD BX,8
MOV SI,0
LOOP INPUT   ;三个数据输入完毕，存储真正的值放在第一位
POP DX
MOV AH,09H
INT 21H
MOV CX,0
SUB BX,24
MOV AL,BYTE PTR[BX]

PUSH BX
MOV BL,3
XOR AH,AH
IMUL BL
POP BX
MOV BYTE PTR[BX+24],AL ;将X的值存放于RESULT的第0位
MOV AL,[BX+8]
ADD AL,0
JS ABSY
```

INY:

```
PUSH BX
MOV BL,02H
XOR AH,AH
MUL BL
POP BX
MOV [BX+25],AL ;将|Y|的值存放于RESULT的第1位
MOV AL,BYTE PTR[BX+16]
ADD AL,0
JS ABSZ
```

INZ:

```
MOV [BX+26],AL ;将|Z|的值存放于RESULT的第2位
```

```

ADD BX,24
MOV AH,0
MOV AL,[BX]
SUB AL,[BX+1]
ADD AL,[BX+2]      ;公式运算
JS DISPLAYNEG;判断结果是否为负数

```

STO:

```

MOV DX,0
MOV AH,0
INC CX
IDIV BP
PUSH DX      ;十进制数的每一位入栈
CMP AX,0
JNZ STO
MOV AH,2

```

DIS:

```

POP DX
ADD DL,30H
INT 21H      ;十进制数的每一位出栈并输出
LOOP DIS

```

```

MOV AX,4C00H
INT 21H

```

MAIN ENDP

SOLO:

```

SUB AL,30H      ;直接处理
JMP S

```

INCHAR:

```

MOV AH,01H
INT 21H
MOV [BX+SI],AL      ;获取单个字符
INC SI
CMP AL,2DH
JZ GETF      ;若是负号则跳转

```

JUG:

```

CMP AL,0DH
JNZ INCHAR
RET      ;判断结束

```

EXC PROC NEAR

```

SUB AL,30H
MUL BP
ADD AL,[BX+SI]
ADD SI,1      ;字符串每一位累加计算数值
LOOP EXC
SUB AL,30H
MOV BX,109
CMP BYTE PTR[BX],1
JZ NEGA      ;若符号位为1，则表明是负数，取补码
RET

```

```

EXC ENDP

GETF:
    MOV BYTE PTR[BX], 30H
    PUSH BX
    MOV BX, 109
    MOV BYTE PTR[BX], 1
    POP BX
    JMP JUG

NEGA:
    NEG AL
    MOV BYTE PTR[BX], 0    ;取补码，符号位置0
    RET

ABSY:
    NEG AL                ;取绝对值
    JMP INY

ABSZ:
    NEG AL                ;取绝对值
    JMP INZ

DISPLAYNEG:
    NEG AL
    PUSH AX
    MOV DL, '-'
    MOV AH, 02H
    INT 21H              ;显示负数
    POP AX
    JMP STO

CODE ENDS
    END START

```

## 运行结果

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file: 5.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 463k

D:\>if exist 5.OBJ tlink /v/3 5.obj >X:\LINK.LOG

Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
Warning: No stack

D:\>5.exe
Enter 3 numbers
Please input X:-7
Please input Y:2
Please input Z:5
3X-2|Y|+|Z|= -20
(END)Here is the end of the program's output

Do you need to keep the DOSBox [Y,N]?
```

## 算法分析

- 1.多位的数字利用选择结构处理成为数值形式
- 2.利用取补码置零，对绝对值进行处理
- 3.根据运算结果决定是否需要输出 ' - '

## 题目六

### 原题

题目：分类统计字符个数

内容：程序接受用户从键盘输入的一行字符（字符个数不超过 80 个字符，该字符串以回车符结束），并按字母、数字及其他字符分

类统计个数，然后将相应的结果存放于 letter、digit 和 other 中，并在显示器上显示如下信息：

The counted result of the program:

letter:xx

digit:xx

other:xx

xx表示一个十进制数

# 源代码

```
DATA SEGMENT
    STRING1 DB 'Input your string:$';指示说明
    STRING2 DB 'Digit:$';指示说明
    STRING3 DB 'Letter:$';指示说明
    STRING4 DB 'Other:$';指示说明
    DIGIT DB 0;用于存放数字数量
    LETTER DB 0;用于存放字母数量
    OTHER DB 0;用于存放其他字符数量
DATA ENDS

STA SEGMENT;堆栈段
    DB 20 DUP(?)
STA ENDS

CODE SEGMENT
    ASSUME DS:DATA,CS:CODE,SS:STA;段地址分配伪指令
MAIN PROC FAR;定义主函数
START:
    MOV AX,DATA
    MOV DS,AX;数据段赋值
    LEA DX,STRING1
    MOV AH,9
    INT 21H;输出指示说明STRING1
    MOV CX,80;最大输入长度80
P1:
    MOV AH,1
    INT 21H;读取一个字符
    CMP AL,0DH;与回车(0DH)比较,结束输入
    JZ OVER;跳转至结束程序
    CMP AL,30H;与0(30H)比较
    JB OTHERS;比0(30H)小,OTHER+1
    CMP AL,39H;与9(39H)比较
    JA P2;进入P2,准备进行字母判断
    JMP DIGITS;若未跳转,说明字符在0-9,DIGIT+1

P2:
    CMP AL,41H;与A(41H)比较
    JB OTHERS;小于A(41H),OTHER+1
    CMP AL,5AH;与Z(5AH)比较
    JA P3;比Z(5AH)大,跳转至P3,进行小写字母判断
    JMP LETTERS;若未跳转,说明字符在A-Z,LETTER+1;

P3:
    CMP AL,61H;与a(61H)比较
    JB OTHERS;小于a(61H),OTHER+1
    CMP AL,7AH;与z(7AH)比较
    JA OTHERS;比z(7AH)大,OTHER+1
    JMP LETTERS;若未跳转,说明字符在a-z,LETTER+1

DIGITS:
    INC DIGIT
    JMP L

OTHERS:
    INC OTHER
```



JMP L

LETTERS:

INC LETTER

JMP L

L:

LOOP P1;循环进行判断

OVER:

LEA DX,STRING2

MOV AH,9

INT 21H;输出指示信息

XOR AX,AX;清空AX

MOV AL,DIGIT

CALL PRINT;打印DIGIT

CALL RE;输出回车

LEA DX,STRING3

MOV AH,9

INT 21H;输出指示信息

XOR AX,AX

MOV AL,LETTER

CALL PRINT;打印LETTER

CALL RE;输出回车

LEA DX,STRING4

MOV AH,9

INT 21H;输出指示信息

XOR AX,AX

MOV AL,OTHER

CALL PRINT;打印OTHER

MOV AX,4C00H

INT 21H

MAIN ENDP;主函数结束

RE PROC NEAR

MOV AH,2

MOV DL,0AH

INT 21H;回车

MOV AH,2

MOV AL,0DH

INT 21H;换行

RET

RE ENDP

PRINT PROC NEAR;输出一个二位十进制数

MOV DL,0AH

DIV DL

MOV BX,AX

MOV DL,BL

ADD DL,30H

MOV AH,02H

INT 21H

MOV DL,BH

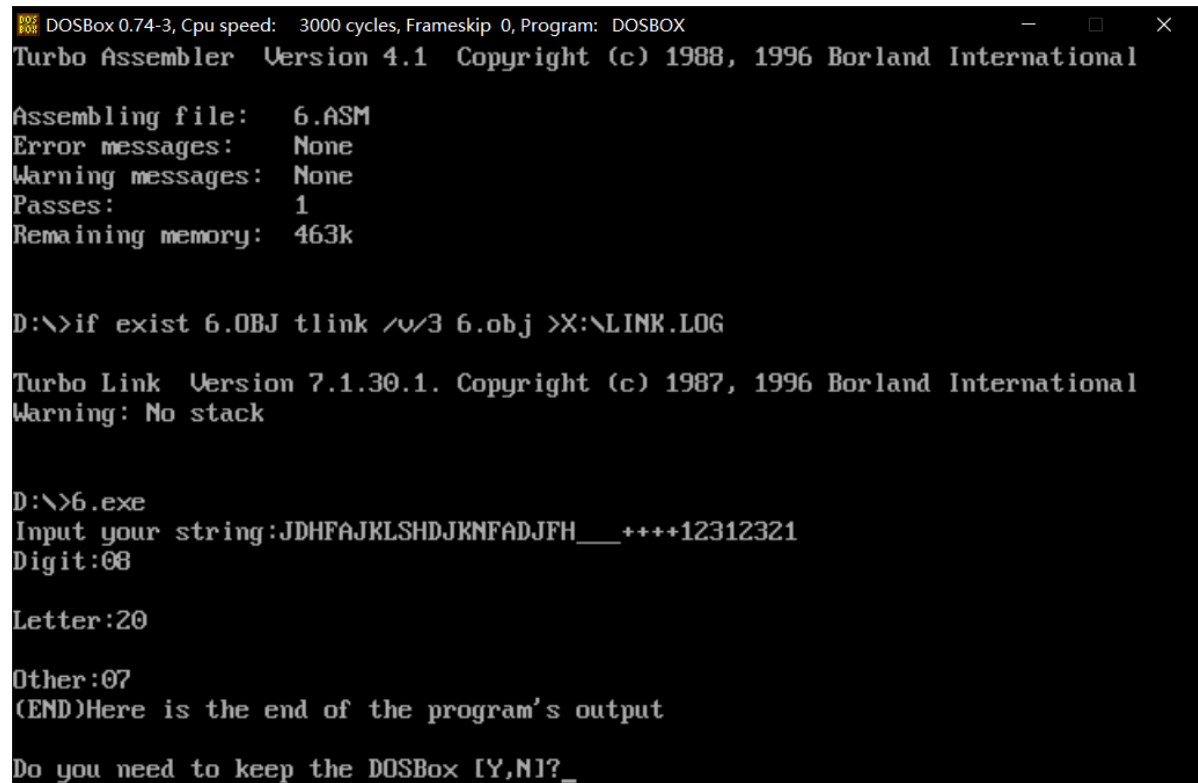
```
ADD DL,30H
INT 21H
RET

PRINT ENDP

CODE ENDS

END START
```

## 运行结果



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Turbo Assembler Version 4.1 Copyright (c) 1988, 1996 Borland International

Assembling file: 6.ASM
Error messages: None
Warning messages: None
Passes: 1
Remaining memory: 463k

D:\>if exist 6.OBJ tlink /v/3 6.obj >X:\LINK.LOG

Turbo Link Version 7.1.30.1. Copyright (c) 1987, 1996 Borland International
Warning: No stack

D:\>6.exe
Input your string:JDHF AJKLSHDJKNFADJFH____++12312321
Digit:08

Letter:20

Other:07
(END)Here is the end of the program's output

Do you need to keep the DOSBox [Y,N]?_
```

## 算法分析

- 1.每次输入一个字符就进行一次判断，对应存储单元数值+1即可
- 2.通过循环选择结构实现统计功能