哈尔滨工业大学（深圳）

实践教学报告

学　院: 计算机科学与技术学院

题　目: 实验二：分支程序与循环程序设计

姓　名: 李秋阳

学　号: 180110527

专　业: 计算机科学与技术

日　期: 2019 年 10 月 17 日

一、问题描述

必做题：

题目：试编写一段程序，找出首地址为 number 的 10 个字类型的数字中的正奇数并求和,结果放入 result 中，并把它在屏幕上显示出来。

通过对所有数字进行正负性判断和奇偶性判断找出正奇数并求和，然后输出

选做题 1：

题目：实现选择排序算法，完成对无序数组从小到大的排序。

结合条件转移语句设计汇编语言的选择排序算法，完成排序

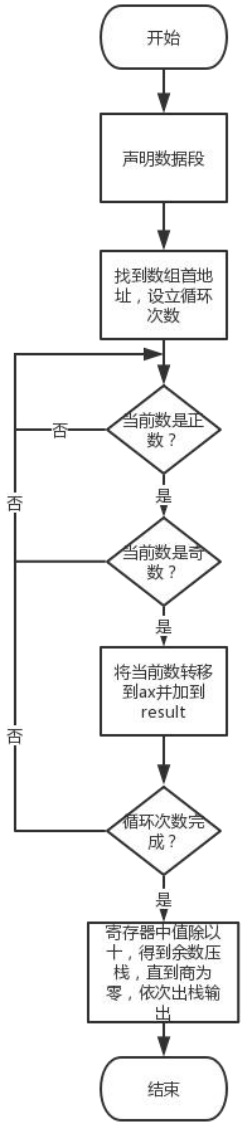
选做题 2：

题目：分类统计字符个数，自定义一个字符串(字符串长度大于 20，$表示字符串结束)，按字母、数字以及其他字符三类进行分类计数，将计数结果分别存储到以 letter、digit 和 others 为名的存储单元中

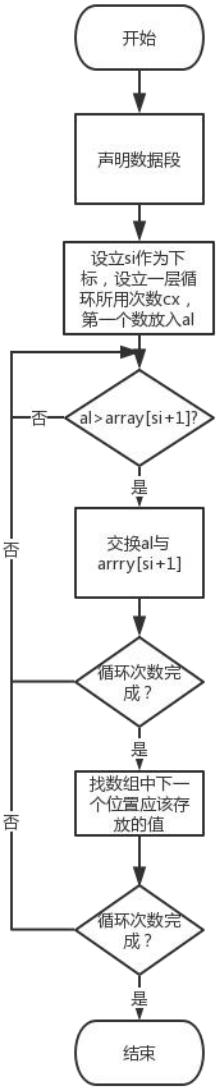
通过对所有字符判断其 ASCII 码处于哪个区间，分类计数，统计个数

二、解决方案

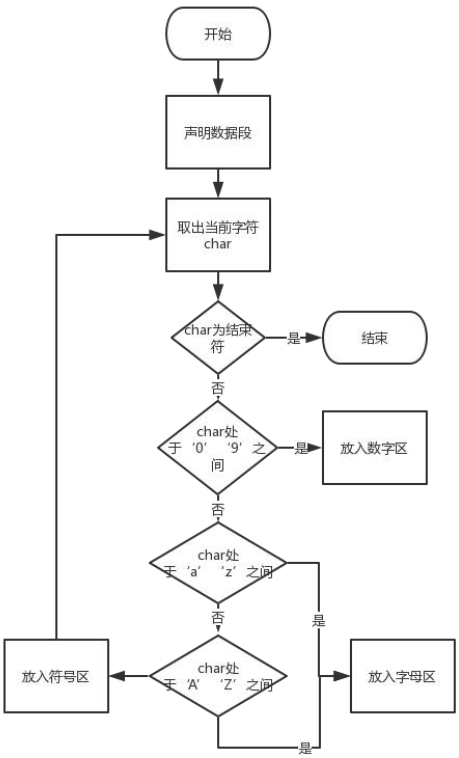
必做题：将原始数据存放于一个数组当中，找到数组首地址，设立循环次数，从首地址开始遍历数组，对其中的每一位进行判断，跳过负数和偶数，将正奇数转移到 ax 寄存器中并求和。将寄存器中的值除以 10，得到当前最低位（余数）后压栈，对所得的商重复操作，直到最后商为 0 的时候结束循环。依次弹出栈顶元素，加上 30h 后转换为 ASCII 码，送 dl，调用 21h 中断输出字符。



选做题 1：选择排序思想即为每一次从待排序的数据元素中选出最小的一个元素，存放在序列的起始位置，直到全部待排序的数据元素排完。通过设立内外两层循环，结合条件转移语句，每次将找到的最小值放到应在的数组位置。



选做题 2：运用基址+变址寻址的方法对字符串进行扫描，对于每一个字符，先进行是否为数字的判断，对 ASCII 码进行比较，分两次比较，确保在区间范围内。若在区间范围外，则再进行是否为大写字母的判断，若仍然在区间范围外，则再进行小写字母的判断，最终若都不满足上述情况，则属于其他符号。



三、具体实现

必做题

```
data Segment
number dw 24,13,-5,7,-101,28,46,77,100,3
result dw ?
data ends
stacks segment
    stack dw 100 dup(?)
stacks ends
program Segment
main proc far
assume ds:data,cs:program,ss:stacks
s:
    mov ax,data
    mov ds,ax
    mov ax,stacks
    mov ss,ax
    mov bx,0

start:
    cmp bx,20
    jz begin
    cmp number[bx],0
    jng X
    test number[bx],01h
    jz X
    mov ax,number[bx]
    add result,ax
X:
    add bx,2
    jmp start
begin:
    mov bx,10
    mov ax,result
    mov cx,1

    call print
    ret
main endp

print proc near
divs:
    mov dx,0
    div bx
    push dx
    cmp ax,0
    jz show
    inc cx
    jmp divs
show:
    pop dx
    add dl,30h
    mov ah,2
    int 21h
    loop show
print endp

exit:
    mov ax, 4c00h
    int 21h

program ends
end s
```

结果

```
D:\>Podd
100
```

```
D:\>debug Podd.exe
-u
0779:0000 B86A07        MOV     AX,076A
0779:0003 8ED8          MOV     DS,AX
0779:0005 B86C07        MOV     AX,076C
0779:0008 8ED0          MOV     SS,AX
0779:000A BB0000        MOV     BX,0000
0779:000D 83FB14        CMP     BX,+14
0779:0010 741C          JZ      002E
0779:0012 83BF000000    CMP     WORD PTR [BX+0000],+00
0779:0017 7E10          JLE     0029
0779:0019 F787000000100 TEST    WORD PTR [BX+0000],0001
0779:001F 7408          JZ      0029
```

```
-u
0779:0021 8B870000      MOV     AX,[BX+0000]
0779:0025 01061400      ADD     [0014],AX
0779:0029 83C302        ADD     BX,+02
0779:002C EBDF          JMP     000D
0779:002E BB0A00        MOV     BX,000A
0779:0031 A11400        MOV     AX,[0014]
0779:0034 B90100        MOV     CX,0001
0779:0037 E80100        CALL    003B
0779:003A CB            RETF
0779:003B BA0000        MOV     DX,0000
0779:003E F7F3          DIV     BX
0779:0040 52            PUSH    DX
```

```
-u
0779:0041 3D0000        CMP     AX,0000
0779:0044 7403          JZ      0049
0779:0046 41            INC     CX
0779:0047 EBF2          JMP     003B
0779:0049 5A            POP     DX
0779:004A 80C230        ADD     DL,30
0779:004D B402          MOV     AH,02
0779:004F CD21          INT     21
0779:0051 E2F6          LOOP    0049
0779:0053 B8004C        MOV     AX,4C00
0779:0056 CD21          INT     21
0779:0058 5D            POP     BP
0779:0059 C3            RET
0779:005A 55            PUSH    BP
0779:005B 8BEC          MOV     BP,SP
0779:005D 81EC8600      SUB     SP,0086
```

```
-g53
100
AX=0230  BX=000A  CX=0000  DX=0030  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=076C  CS=0779  IP=0053   NV UP EI PL NZ NA PE NC
0779:0053 B8004C        MOV     AX,4C00
-d0     计算结果
076A:0000  18 00 0D 00 FB FF 07 00-9B FF 1C 00 2E 00 4D 00   ..............M.
076A:0010  64 00 03 00 64 00 00 00-00 00 00 00 00 00 00 00   d...d...........
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

选做题一

```
data segment
    array db 12,23,2,4,3,9,8,34,21,44,55,66,77,11,90
    count db $-array ;数组长度
data ends
code segment
    assume cs:code,ds:data
start:
    mov ax,data
    mov ds,ax

    mov ax,0
    mov cx,0
    mov cl,count
    dec cx;比较n-1次
    mov bx,0
loop1:
    push cx
    mov si,bx;内循环下标
    mov al,array[si]
loop2:
    cmp al,array[si+1]
    jg next;设为最小值，交换
    jmp done
next:
    xchg al,array[si+1];暂存
done:
    inc si
    loop loop2
    mov array[bx],al;得到一个最小值
    inc bx
    pop cx
    loop loop1
    mov ax,4c00h
    int 21h
    code ends
    end  start
```

结果

```
D:\>debug sort.exe
-u
076B:0000 B86A07        MOV     AX,076A
076B:0003 8ED8          MOV     DS,AX
076B:0005 B80000        MOV     AX,0000
076B:0008 B90000        MOV     CX,0000
076B:000B 8A0E0F00      MOV     CL,[000F]
076B:000F 49            DEC     CX
076B:0010 BB0000        MOV     BX,0000
076B:0013 51            PUSH    CX
076B:0014 8BF3          MOV     SI,BX
076B:0016 8A840000      MOV     AL,[SI+0000]
076B:001A 3A840100      CMP     AL,[SI+0001]
076B:001E 7F03          JG      0023
```

```
-u
076B:0020 EB05          JMP     0027
076B:0022 90            NOP
076B:0023 86840100      XCHG    AL,[SI+0001]
076B:0027 46            INC     SI
076B:0028 E2F0          LOOP    001A
076B:002A 88870000      MOV     [BX+0000],AL
076B:002E 43            INC     BX
076B:002F 59            POP     CX
076B:0030 E2E1          LOOP    0013
076B:0032 B8004C        MOV     AX,4C00
076B:0035 CD21          INT     21
076B:0037 C404          LES     AX,[SI]
076B:0039 50            PUSH    AX
076B:003A E87B0E        CALL    0EB8
076B:003D 83C404        ADD     SP,+04
```

```
-g32

AX=004D  BX=000E  CX=0000  DX=0000  SP=0000  BP=0000  SI=000E  DI=0000
DS=076A  ES=075A  SS=0769  CS=076B  IP=0032   NV UP EI PL NZ NA PO CY
076B:0032 B8004C        MOV     AX,4C00
-d0
076A:0000  02 03 04 08 09 0B 0C 15-17 22 2C 37 42 4D 5A 0F   .........",7BMZ.
076A:0010  B8 6A 07 8E D8 B8 00 00-B9 00 00 8A 0E 0F 00 49   .j.............I
076A:0020  BB 00 00 51 8B F3 8A 84-00 00 3A 84 01 00 7F 03   ...Q......:.....
076A:0030  EB 05 90 86 84 01 00 46-E2 F0 8B 87 00 00 43 59   .......F......CY
076A:0040  E2 E1 B8 00 4C CD 21 C4-04 50 E8 7B 0E 83 C4 04   ....L.!..P.{....
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&.G.*
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
```

选做题二

```
assume cs:code,ds:data
data segment
    string db 'abc#1L0*9012~\/<>jky6932','$'
    numbers db 0
    words db 0
    others db 0
data ends

code segment
start:
    mov ax,data
    mov ds,ax
    mov si,0
s:
    mov al,string[si]
    cmp al,'$'
    je ok
    cmp al,'0'
    jb sother
    cmp al,'9'
    ja another
    mov bl,numbers
    inc bl
    mov numbers,bl
    inc si
    jmp s
another:
    cmp al,'A'
    jb sother
    cmp al,'Z'
    ja ananother
    mov bl,words
    inc bl
    mov words,bl
    inc si
    jmp s
ananother:
    cmp al,'a'
    jb sother
    cmp al,'z'
    ja sother
    mov bl,words
    inc bl
    mov words,bl
    inc si
    jmp s
sother:
    mov bl,others
    inc bl
    mov others,bl
    inc si
    jmp s
ok:
    mov ax,4c00h
    int 21h
code ends
end start
```

结果

```
D:\>debug censor.exe
-u
076C:0000 B86A07        MOV     AX,076A
076C:0003 8ED8          MOV     DS,AX
076C:0005 BE0000        MOV     SI,0000
076C:0008 8A840000      MOV     AL,[SI+0000]
076C:000C 3C24          CMP     AL,24
076C:000E 744C          JZ      005C
076C:0010 3C30          CMP     AL,30
076C:0012 723B          JB      004F
076C:0014 3C39          CMP     AL,39
076C:0016 770D          JA      0025
076C:0018 8A1E1900      MOV     BL,[0019]
076C:001C FEC3          INC     BL
076C:001E 881E1900      MOV     [0019],BL
```

```
-u
076C:0022 46            INC     SI
076C:0023 EBE3          JMP     0008
076C:0025 3C41          CMP     AL,41
076C:0027 7226          JB      004F
076C:0029 3C5A          CMP     AL,5A
076C:002B 770D          JA      003A
076C:002D 8A1E1A00      MOV     BL,[001A]
076C:0031 FEC3          INC     BL
076C:0033 881E1A00      MOV     [001A],BL
076C:0037 46            INC     SI
076C:0038 EBCE          JMP     0008
076C:003A 3C61          CMP     AL,61
076C:003C 7211          JB      004F
076C:003E 3C7A          CMP     AL,7A
076C:0040 770D          JA      004F
```

```
-u
076C:0042 8A1E1A00      MOV     BL,[001A]
076C:0046 FEC3          INC     BL
076C:0048 881E1A00      MOV     [001A],BL
076C:004C 46            INC     SI
076C:004D EBB9          JMP     0008
076C:004F 8A1E1B00      MOV     BL,[001B]
076C:0053 FEC3          INC     BL
076C:0055 881E1B00      MOV     [001B],BL
076C:0059 46            INC     SI
076C:005A EBAC          JMP     0008
076C:005C B8004C        MOV     AX,4C00
076C:005F CD21          INT     21
076C:0061 FE81E6FF      INC     BYTE PTR [BX+DI+FFE6]
```

```
-g5c

AX=0724  BX=000A  CX=0081  DX=0000  SP=0000  BP=0000  SI=0018  DI=0000
DS=076A  ES=075A  SS=0769  CS=076C  IP=005C   NV UP EI PL ZR NA PE NC
076C:005C B8004C        MOV     AX,4C00
-d 076A:0
076A:0000  61 62 63 23 31 4C 30 2A-39 30 31 32 7E 5C 2F 3C   abc#1L0*9012~\/<
076A:0010  3E 6A 6B 79 36 39 33 32-24 0A 02 02 00 00 00 00   >jky6932$.......
076A:0020  B8 6A 07 8E D8 BE 00 00-8A 84 00 00 3C 24 74 4C   .j..........<$tL
076A:0030  3C 30 72 3B 3C 39 77 0D-8A 1E 19 00 FE C3 88 1E   <0r;<9w.........
076A:0040  19 00 46 EB E3 3C 41 72-26 3C 5A 77 0D 8A 1E 1A   ..F..<Ar&<Zw....
076A:0050  00 FE C3 88 1E 1A 00 46-EB CE 3C 61 72 11 3C 7A   .......F..<ar.<z
076A:0060  77 0D 8A 1E 1A 00 FE C3-88 1E 1A 00 46 EB B9 8A   w...........F...
076A:0070  1E 1B 00 FE C3 88 1E 1B-00 46 EB AC B8 00 4C CD   .........F....L.
```

四、总结

在本次实验中，我们主要学习了各种跳转指令的使用。例如，条件跳转指令和无条件转移指令。在不同的情况下，需要选择不同的转移指令，同时，对标志寄存器的判断也是很重要的。在实验过程中，需要灵活的使用逻辑移位指令代替乘除运算，使用 test 指令进行奇偶判断，利用标志寄存器进行条件跳转，将会极大提高代码的简洁程度、节省寄存器的使用，并提高程序的运行效率。