

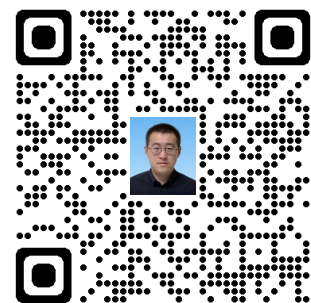


東北大學
Northeastern University

汇编语言程序设计

主讲：刘松冉
单位：东北大学计算机学院
智慧系统国际联合实验室

联系方式：liusongran@cse.neu.edu.cn
个人主页：<http://faculty.neu.edu.cn/liusongran>
<https://liusongran.github.io/>



第十一章 列表与字符串操作

一. 列表处理

二. 记录与结构

三. 列表的种类及其运算

四. 字符串处理



一. 列表处理

► **列表的定义：**列表即线性表，是指一组有限的数据的集合，它们前后相接，没有分支。
可表示为：

$$a_1, a_2, a_3, a_4, \dots, a_i, \dots, a_n$$

例如：一星期中的七天，依次为：星期一，星期二，星期三，星期四，星期五，星期六，星期日。这是简单的列表，其中星期一，星期二，...，星期日称为列表的元素。

再如下表中的学生成绩表，就是复杂些的列表。

学生成绩表

学号	姓名	计算机	外语	数学	物理	体育	总成绩	平均成绩
001	黎明	95	89	98	93	80		
002	宋妙然	95	90	95	90	90		
003	李昕	94	93	97	90	90		
...		
895	章山	78	67	89	76	90		

一. 列表处理

其中每个学生的成绩在表中占一行，每行的信息说明某个学生五门课的学习成绩、总成绩及平均成绩。表中的每一行称为一个**元素**（或记录），元素中的学号、姓名等称为**数据项或字段**。

一般地讲，列表是由一组数据组成的，在不同的情况下，一个数据元素的具体含义可以是不同的。总的来说，列表有如下结构特征：

- 1) 列表是数据元素的一个有限序列。表的长度可定义为列表中数据元素的个数 N ，当 $N=0$ 时，为空表
- 2) 数据元素在列表中的位置只取决于它们自己的序号，数据元素之间的相对位置是线性的

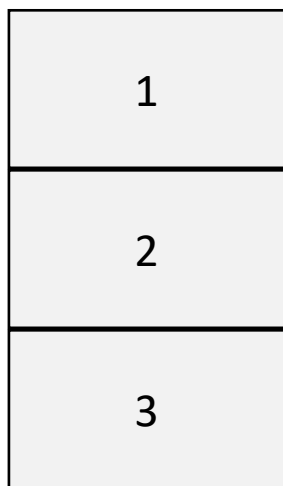
学生成绩表

学号	姓名	计算机	外语	数学	物理	体育	总成绩	平均成绩
001	黎明	95	89	98	93	80		
002	宋妙然	95	90	95	90	90		
003	李昕	94	93	97	90	90		
...		
895	章山	78	67	89	76	90		

一. 列表处理

► 列表的存储结构

- 1) **顺序存储**：即用一组连续的存储单元依次存储列表中的各元素。这种方法不仅简单，而且访问某个元素很方便，可以直接计算找出。但是要插入、删除或排序操作，将引起元素的大量移动。
- 2) **链式存储**：除了保存元素的值外，还必须设有一个指示下一个元素存放地址的信息。对这种形式存储的列表进行插入，删除不需要移动其它信息，但却是以牺牲存储空间为代价的。



类比理解：数组



类比理解：双向链表

一. 列表处理

▶ 例子：

以顺序存储形式存储上述的学生成绩表，则应先确定元素的长度，即一个元素所占用的字节数。元素长度是元素所包含的数据项的长度之和。比如，学号字段用二进制表示，最大学号为1000，那这个数据项就需要两个字节；姓名用汉字代码存储，每个名字最多四个汉字，每个汉字占用两个字节（机内码），共8个字节；每科成绩和平均成绩最大值为100，用二进制形式存储，占用一个字节；总成绩最大值为500，用二进制形式存储，占用2个字节；各字段占用字节总数为18。设学生成绩表的首地址为STUDENT，在计算机内存储格式如图。

STUDENT + 0		学号
+ 1		
+ 2		学生姓名
:		
+ 9		
+ 10		
+ 11		外语
+ 12		数学
+ 13		物理
+ 14		体育
+ 15		总成绩
+ 16		
+ 17		平均成绩

二. 记录与结构 — 记录

1. 记录

当列表元素包含的数据项较少，每个数据项用几位二进制数就可以表示，列表元素的总长度不超出8位或16位二进制数时，就可以把一个数据项放在一个字节或两个字节中。这时，可以采用**记录数据**语句来定义。

1) 记录定义伪指令 **RECORD**

格式： **RN** **RECORD** FN1 : WT1[, FN2 : WT2 , ...]

说明： RN为记录名，是用户自定义标识符号，FN1，FN2，...为记录的字段名，是用户自定义的标识符号；WT1，WT2为字段宽度，是1~16间的常数。如果字段的总宽度小于等于8，汇编程序将用一个字节表示记录；如果字段的总宽度大于8，小于等于16，汇编程序将用2个字节表示记录。汇编程序把所定义的字段与字节或字的最低有效位对齐。

注意：记录定义伪指令并不分配内存单元，它只告诉汇编程序记录名及记录中各字段的名称、位置和记录的长度。当记录定义后，在程序中可以直接引用记录名，设置记录变量，分配内存空间。

二. 记录与结构 — 记录

▶ 2. 预置记录及存储分配

格式：[SN] **RN** <[EXP][, EXP][, ...]>

说明：SN为记录变量名，是用户自定义的标识符号，表示记录变量的存储分配的第一个字节或字的符号地址，是任选项。RN是RECORD伪指令定义的记录名；尖括号内的EXP为表达式，用以初始化记录字段的数据，它可以是常数、字符或缺省。缺省值为0。

例：定义记录和初始化记录

MODE	RECORD	A : 4 , B : 5 , C : 7
AS_MODE	MODE	<2 , , 17>
RS_MODE	MODE	<9 , 15 , 26>
CS_MODE	MODE	< , 8 , >

存储单元分配情况：

AS_MODE	0010000000010001
RS_MODE	1001011110011010
CS_MODE	0000010000000000

二. 记录与结构 — 记录

▶ 3. 记录的运算 – 记录的宽度运算

格式： **WIDTH** RN (或FN)

说明： RN为记录名，FN为字段名。运算的结果为该记录或字段的宽度（二进制位数）。

例： W1 EQU WIDTH MODE
 MOV AH, WIDTH A
则W1=10H，AH=4。

定义记录和初始化记录

MODE	RECORD	A : 4 , B : 5 , C : 7
AS_MODE	MODE	<2 , , 17>
RS_MODE	MODE	<9 , 15 , 26>
CS_MODE	MODE	< , 8 , >

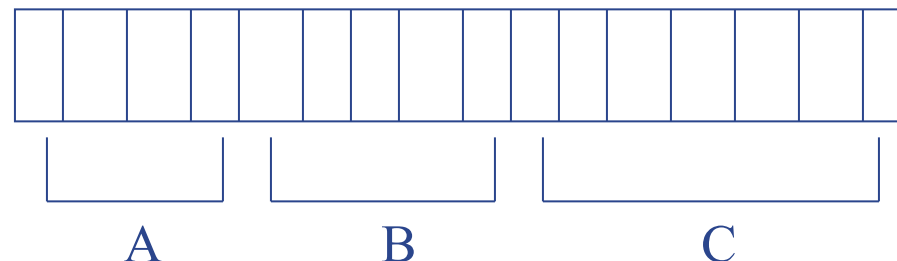
二. 记录与结构 — 记录

▶ 3. 记录的运算 – 位移值运算

说明：在语句中可以把字段名直接作为操作数使用，其意义为该字段右移到所在记录的最右边的移位次数。它也表示该字段在记录中的位置（从右到左，范围0~15）。

例： MOV AL, A
 MOV BL, B
 MOV CL, C

则AL, BL, CL值分别为12，7，0。



定义记录和初始化记录

MODE	RECORD	A : 4 , B : 5 , C : 7
AS_MODE	MODE	<2 , , 17>
RS_MODE	MODE	<9 , 15 , 26>
CS_MODE	MODE	< , 8 , >

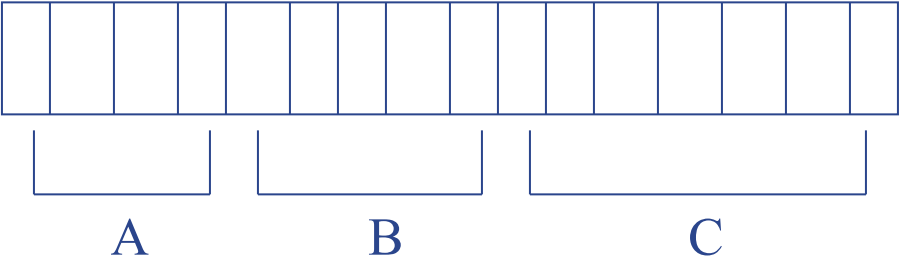
二. 记录与结构 — 记录

▶ 3. 记录的运算 – 屏蔽运算

格式：**MASK** FN

说明：FN为记录字段名，MASK运算的结果为该记录字段在记录中的屏蔽码，即该字段各位均为“1”，而记录中其它各位均为“0”的代码。

例：
MASK A ;0F000H
MASK B ;0F80H
MASK C ;007FH



定义记录和初始化记录

MODE	RECORD	A : 4 , B : 5 , C : 7
AS_MODE	MODE	<2 , , 17>
RS_MODE	MODE	<9 , 15 , 26>
CS_MODE	MODE	< , 8 , >

二. 记录与结构 — 记录

4. 应用举例

例1-part1：现有200名学生，把学生的基本情况建立一个档案，格式如表11.2所示。

- 学号：1~200，8bit；
- 性别：1bit；
- 年龄：15~30，5bit；
- 政治面貌：2bit。

则可以用16位二进制位表示每个元素，可用记录表示每个元素，定义如下：

STUDNT

S_RCD

RECORD

STUDNT

STUDNT

STUDNT

:

STUDNT

NUMB : 8 , SEX : 1 , AGE : 5 , PL : 2

<1,1,18,1>

<2,0,19,1>

<3,1,20,3>

:

<200,1,21,2>

表11.2		学生档案	
学 号	性 别	年 龄	政治面貌
001	男	18	团员
002	女	19	团员
003	男	20	党员
:	:	:	:
200	男	21	预备党员

二. 记录与结构 — 记录

4. 应用举例

例1-part1：现有200名学生，把学生的基本情况建立一个档案，格式如表11.2所示。

- 学号：1~200，8bit；
- 性别：1bit；
- 年龄：15~30，5bit；
- 政治面貌：2bit。

则可以用16位二进制位表示每个元素，可用记录表示每个元素，定义如下：

STUDENT	RECORD	NUMB：8，SEX：1，AGE：5，PL：2
S_RCD	STUDENT	<1,1,18,1>
	STUDENT	<2,0,19,1>
	STUDENT	<3,1,20,3>
	:	
	STUDENT	<200,1,21,2>

上述记录在内存中的存储形式如下：

S_RCD+0	11001001
+1	00000001
+2	01001101
+3	00000010
+4	11010011
+5	00000011

二. 记录与结构 — 记录

▶ 4. 应用举例

例1-part2：利用上面定义的记录，编制程序统计男同学中团员人数，并存入TOTAL单元。

- 学号：1~200，8bit；
- 性别：1bit；
- 年龄：15~30，5bit；
- 政治面貌：2bit。

解题步骤：

- 1) 设置记录地址指针；
- 2) 设置循环计数器的初值；
- 3) 设置计数器记录“男学生团员”人数；
- 4) 读取记录；
- 5) 是男学生吗？不是，转8；
- 6) 是团员吗？不是，转8；
- 7) 计数器增1；
- 8) 统计完？未完，转4；
- 9) 保存统计值；
- 10) 结束。

二. 记录与结构 — 记录

4. 应用举例

例1-part2：利用上面定义的记录，编制程序统计男同学中团员人数，并存入TOTAL单元。

```
1  DSEG  SEGMENT
2  STUDNT RECORD      N:8,S:1,L:5,T:2
3  SRCD   STUDNT      <1,1,18,1>
4          STUDNT      <2,0,19,1>
5          STUDNT      <3,1,20,3>
6          STUDNT      <4,1,17,0>
7          STUDNT      <5,0,17,0>
8          STUDNT      <200,1,21,2>
9  RLNT   DB          6
10 TOTAL  DB          0
11 DSEG  ENDS
12
13 CSEG  SEGMENT
14          ASSUME CS:CSEG,DS:DSEG
15 START: MOV  AX, DSEG
16          MOV  DS, AX
17          LEA  SI, SRCD
18          MOV  CL, RLNT
19          MOV  DL, 0
20 AGAIN: MOV  AX, [SI]
21          TEST AL, MASK  S
22          JZ   CHCNT
23          AND  AX, MASK  T
24          CMP  AX, 01
25          JNZ  CHCNT
26          INC  DL
27 CHCNT: ADD  SI, 2
28          DEC  CL
29          JNZ  AGAIN
30          MOV  TOTAL, DL
31          MOV  AH, 4CH
32          INT  21H
33 CSEG  ENDS
34          END  START
```

解题步骤：

- 1) 设置记录地址指针；
- 2) 设置循环计数器的初值；
- 3) 设置计数器记录“男学生团员”人数；
- 4) 读取记录；
- 5) 是男学生吗？不是，转8；
- 6) 是团员吗？不是，转8；
- 7) 计数器增1；
- 8) 统计完？未完，转4；
- 9) 保存统计值；
- 10) 结束。

二. 记录与结构 — 结构

1. 结构定义

记录伪指令只能用于表示两字节以内的数据结构，当数据元素的长度超出字范围时，则可以用结构伪指令加以定义。

结构定义伪指令 **STRUC**

格式：

STN **STRUC**

FN PSD EXP1 [, EXP2 , ...]

: **:**

: **:**

STN ENDS

说明： STN是用户定义的结构名字，在源程序中必须是唯一的；FN是用户定义的结构成员名；PSD为数据定义伪指令DB，DW，DD，DQ或DT，但不分配内存单元，EXP为相应字段的缺省值，当分配结构的内存时，如相应字段无初始值时，用此缺省值代替。当字段只有一项缺省值时，此缺省值可用另外的值代替，称为可替代型的；当字段有两个以上缺省值时，不可用其它值代替，称为不可替代型的。

二. 记录与结构 — 结构

2. 预置结构及存储分配

设置结构变量和设置记录变量类似。当结构定义后，在程序中可以直接引用结构名，设置结构变量，分配内存空间。

预置结构变量的语句格式如下：

格式：**[SN] STN EXP**

说明：其中SN为结构变量名，是用户自定义标识符号，它同样有三个属性：段、偏移量和类型属性。STN是STRUC/ENDS伪指令定义的结构名。

```
1  ;例：定义和预置结构
2  STR_D  STRUC
3  FN1    DB  0           ;可替代型
4  FN2    DB  5,7         ;不可替代型
5  FN3    DB  3 DUP(?)    ;不可替代型
6  FN4    DB  'ABCDEE'    ;可替代型
7  STR_D  ENDS
8
9  DATA  SEGMENT
10 STR1    STR_D  <1,,, 'ABCD'>
11 STR2    STR_D  <2,,, 'AAAAAA'>
12 STR3    STR_D  <,,, 'SSSSSQQQQQ'>
13 DATA  ENDS
```

当结构定义中的字段缺省值是字符串常数时，可以用另一字符串代替，如果替代字符串比原字符串短（如STR1），则用空格填充右边空缺字符；如果替代字符串比原字符串长（如STR3），则截去多余字符。

二. 记录与结构 — 结构

▶ 3. 对结构变量及其字段的访问

结构变量与一般变量一样，可作为一条指令的操作数。当TYPE运算符作用于结构变量前时，表达式的值为结构变量所占的字节数。

在编制程序时，可以引用结构中的字段，其格式为：

格式： **结构变量名.字段名**

如： MOV AL, STR1.FN1 ; AL = 1

```
1  ;例：定义和预置结构
2  STR_D   STRUC
3  FN1     DB  0           ;可替代型
4  FN2     DB  5,7         ;不可替代型
5  FN3     DB  3 DUP(?)   ;不可替代型
6  FN4     DB  'ABCDEE'   ;可替代型
7  STR_D   ENDS
8
9  DATA   SEGMENT
10 STR1    STR_D   <1,,, 'ABCD'>
11 STR2    STR_D   <2,,, 'AAAAAA'>
12 STR3    STR_D   <,,, 'SSSSSQQQQ'>
13 DATA   ENDS
```

二. 记录与结构 — 结构

▶ 4. 应用举例

建立学生成绩表（见表11.1），并计算学生的总成绩和平均成绩，添入相应的数据项中，并统计平均成绩在85分以上的人数送入TOTAL单元。

解题步骤：

- 1) 设置数据段指针；
- 2) 取表首地址；
- 3) 设置计数器记录平均85分以上的人数；
- 4) 设置循环计数值；
- 5) 读取元素中各成绩项求出总成绩；
- 6) 计算平均成绩，存入表中；
- 7) 平均成绩在85分以上吗？不是，转9；
- 8) 累计和计数器加1；
- 9) 统计完？未完，转5；
- 10) 保存统计结果；
- 11) 结束。

解题步骤：

- 1) 设置数据段指针；
- 2) 取表首地址；
- 3) 设置计数器记录平均85分以上的人数；
- 4) 设置循环计数值；
- 5) 读取元素中各成绩项求出总成绩；
- 6) 计算平均成绩，存入表中；
- 7) 平均成绩在85分以上吗？不是，转9；
- 8) 累计和计数器加1；
- 9) 统计完？未完，转5；
- 10) 保存统计结果；
- 11) 结束。

```
1  DSEG      SEGMENT
2
3  STULST    STRUC
4  SNUB      DW      0
5  SNAM      DB      '      ';8个空格
6  SCMPT     DB      0
7  SFLAG     DB      0
8  SMATH     DB      0
9  SPHCS     DB      0
10 SPHLY     DB      0
11 TOTL      DW      0
12 SAVER     DB      0
13 STULST    ENDS
14
15 CLNT      DB      5
16 ECOUNT    DW      6
17 S001      STULST  <1, 'JERRY',95,89,98,93,80,,>
18 S002      STULST  <2, 'TOM',95,90,95,90,90,,>
19 S003      STULST  <3, 'ASK',94,93,97,90,90,,>
20 S004      STULST  <4, 'ALEX',78,67,89,76,90,,>
21 S005      STULST  <5, 'ROSE',76,80,90,69,95,,>
22 S006      STULST  <6, 'NSFC',60,68,69,80,90,,>
23 TOTAL     DW      0
24 DSEG      ENDS
25
26 CSEG      SEGMENT
27          ASSUME  CS:CSEG, DS:DSEG
28 START:    MOV    AX, DSEG
29          MOV    DS, AX
30
31
32
33 STAR0:    MOV    CL, CLNT
34
35
36 STAR1:    MOV    DL, [BX][SI]
37
38
39
40
41
42
43
44
45
46
47
48 DN2:      ADD    BX, 18
49
50
51
52
53
54 CSEG      ENDS
55          END    START
```

三. 列表的种类及运算

- ▶ 1. **列表分类**：无序表和有序表。
- 无序表是指整个表元素之间或各个元素的某一数据项之间无任何规律。
 - 有序表则是各个列表元素之间或各个列表元素的某一数据项之间存在着某种规律（由大到小，字典排序等）。

2. 列表的基本运算：

- 1) 求线性表中元素的个数；
- 2) 读表；
- 3) 依赖于元素的某个字段或某些字段的值，按大小递增或递减的次序重新排列列表中的元素；
- 4) 找出列表中的某个元素；
- 5) 确定列表中某个元素的位置；
- 6) 在表的某个位置插入一个新元素；
- 7) 删除线性表中的某个元素。

三. 列表的种类及运算

▶ 3. 列表的插入操作

例：设内存LIST单元开始存放一升序表，元素长度为双字节，表长在LLNT单元。编制程序将ETN单元的元素加到列表中，如果此值不在列表中的话。

- 1) 设置表首地址指针；
- 2) 表长送循环计数器；
- 3) 查表，有待插元素则转10；
- 4) 当前表元素>要插入值吗？是，转7；
- 5) 找完了吗？未完，转3；
- 6) 将待插元素放在表尾，转9；
- 7) 将当前表元素至表尾元素向后移一个位置；
- 8) 将待插元素添入表中；
- 9) 表长加1；
- 10) 结束。

13
28
37
71
100
1001

三. 列表的种类及运算

3. 列表的插入操作

例：设内存LIST单元开始存放一升序表，元素长度为双字节，表长在LLNT单元。编制程序将ETN单元的元素加到列表中，如果此值不在列表中的话。

```
1  DSEG      SEGMENT
2  LLNT      DB      6
3  ETN       DW      60
4  LIST      DW      13,28,37,71,100,1011
5             DW      128 DUP(0)
6  DSEG      ENDS
7
8  CSEG      SEGMENT
9             ASSUME  CS:CSEG,DS:DSEG
10            ASSUME  ES:DSEG
11  START:    MOV     AX,DSEG
12            MOV     DS,AX
13            MOV     ES,AX
14            LEA     SI,LIST
15            MOV     AL,LLNT
16            MOV     CX,ETN
17  INST1:    CMP     CX,[SI]
18            JZ      INST4
19            JC      INST2
20            ADD     SI,2
21
22            DEC     AL
23            JNZ     INST1
24            MOV     [SI],CX
25            JMP     INST3
26  INST2:    MOV     DX,CX
27            MOV     CL,AL
28            MOV     BL,2
29            MUL     BL
30            ADD     SI,AX
31            MOV     DI,SI
32            SUB     SI,2
33            XOR     CH,CH
34            REP     MOVSW
35            MOV     [SI+2],DX
36  INST3:    INC     LLNT
37  INST4:    MOV     AH,4CH
38            INT     21H
39  CSEG      ENDS
40            END     START
```

13
28
37
71
100
1001

三. 列表的种类及运算

4. 列表的删除操作

例：设内存LIST开始存放无序的列表，数据表的长度存放在LLNT单元，表中元素为字节数据，待删的元素存放在ORD单元，如果表中没有待删元素，则置进位。

- 1) 取表首址；
- 2) 表长送计数单元；
- 3) 表中元素=待删元素？是，转7；
- 4) 修改表指针；
- 5) 计数值减1，不为0则转3；
- 6) 置进位位，转10；
- 7) 是表尾？是转9；
- 8) 向前移动表；
- 9) 表长减1；
- 10) 结束。

```
1  √ DSEG      SEGMENT
2  LIST        DB      63,52,11,96,2
3  LLNT        DB      5
4  ORD         DB      11
5  DSEG        ENDS
6
7  √ CSEG      SEGMENT
8          ASSUME  CS:CSEG,DS:DSEG,ES:DSEG
9  √ START:    MOV   AX,DSEG
10         MOV   DS,AX
11         MOV   ES,AX
12         XOR   CH,CH
13         MOV   AL,ORD
14         MOV   CL,LLNT
15         LEA   DI,LIST
16         MOV   SI,DI
17
18
19  DEL1:      ADD   SI,CX
20
21         DEC   SI
22
23         CMP   AL,[DI]
24
25         JZ    DEL2
26
27         INC   DI
28
29         DEC   CL
30
31         JNZ   DEL1
32
33         STC
34
35         JMP   STOP
36
37  DEL2:      CMP   SI,DI
38
39         JZ    DEL3
40
41         MOV   SI,DI
42
43         INC   SI
44
45         CLD
46
47         REP   MOVSB
48
49  DEL3:      DEC   LLNT
50
51  STOP:      MOV   AH,4CH
52
53         INT   21H
54
55  CSEG      ENDS
56
57         END   START
```


三. 列表的种类及运算

5. 列表中的排序 — 交换排序

交换排序也称气泡浮起排序（或冒泡排序）。

第一遍比较：（有N个元素参加，需要比较N-1次，最大值归位）

排序前

第一遍后

9	(9)	7	7	7	7	7	
7	(7)	(9)	9	9	9	9	
18	18	(18)	(18)	3	3	3	
3	3	3	(3)	(18)	6	6	
6	6	6	6	(6)	(18)	13	
13	13	13	13	13	(13)	18	*

三. 列表的种类及运算

► 5. 列表中的排序 — 交换排序 交换排序也称气泡浮起排序（或冒泡排序）。

第二遍比较：（有N-1个元素参加，需要比较N-2次，最大值归位）

排序前

第二遍后

7	(7)	7	7	7	7
9	(9)	(9)	3	3	3
3	3	(3)	(9)	6	6
6	6	6	(6)	(9)	9
13	13	13	13	(13)	13
18	18	18	18	18	18 *

三. 列表的种类及运算

▶ 5. 列表中的排序 — 交换排序 交换排序也称气泡浮起排序（或冒泡排序）。

第三遍比较：（有 $N-2$ 个元素参加，需要比较 $N-3$ 次，最大值归位）

排序前

第三遍后

7	(7)	3	3	7
3	(3)	(7)	6	3
6	6	(6)	(7)	6
9	9	9	(9)	9
13	13	13	13	13
18	18	18	18	18 *

三. 列表的种类及运算

▶ 5. 列表中的排序 — 交换排序 交换排序也称气泡浮起排序（或冒泡排序）。

第四遍比较：（有 $N-3$ 个元素参加，需要比较 $N-4$ 次，最大值归位）

排序前

第四遍后

3	(3)	3	3	
6	(6)	(6)	6	
7	7	(7)	7	
9	9	9	9	
13	13	13	13	
18	18	18	18	*

三. 列表的种类及运算

5. 列表中的排序 — 交换排序

交换排序也称气泡浮起排序（或冒泡排序）。

第五遍比较：（有N-4个元素参加，需要比较N-5次，最大值归位）

排序前		第五遍后	
3	(3)	3	
6	(6)	6	
7	7	7	
9	9	9	
13	13	13	
18	18	18	*

交换排序的算法如下：

- 1) 设置比较遍数计数器初值；
- 2) 设置表首地址指针及遍内比较次数计数器初值；
- 3) 置交换标志初值为0；
- 4) 两相邻元素比较，若次序对则转7；
- 5) 次序不对，两元素交换位置；
- 6) 交换标志置1；
- 7) 元素地址指针指向下一个位置；
- 8) 比较次数计数器值减1；，若不为0则转4；
- 9) 检查交换标志，若为0，则转11；
- 10) 比较遍数计数器值减1，若不为0，则转2；
- 11) 结束。

三. 列表的种类及运算

▶ 5. 列表中的排序 — 交换排序

例子：设内存LIST开始存放一列表，表长在LLNT单元，表中元素为无符号字数据，用交换排序法编制程序将表中元素按升序排列。

```
1  DSEG  SEGMENT
2  LIST  DW      503,87,512,61,908,170,897,275
3  LLNT  DB      8
4  DSEG  ENDS
5
6  CSEG  SEGMENT
7  ASSUME CS:CSEG, DS:DSEG
8  START: MOV     AX,DSEG
9         MOV     DS,AX
10        MOV     CH,LLNT
11        DEC     CH
12  EXCH1: LEA     SI,LIST
13        MOV     CL,CH
14        XOR     BL,BL
15  EXCH2: MOV     AX,[SI]
16        CMP     AX,[SI+2]
```

```
17
18
19
20
21
22  EXCH3: ADD     SI,2
23
24
25
26
27
28
29  STOP:  MOV     AH,4CH
30
31  CSEG  ENDS
32
END      START
```

三. 列表的种类及运算

5. 列表中的排序 — 插入排序

插入排序的基本思想是：对于一个待排序的列表，把其中第一个元素作为已排好序的一个有序子表。然后将列表中的第二个元素插入有序子表，再取第三个元素向有序子表中插入，如此反复，直到整个表元素均已插入到有序子表中为止。

排序前	[25]	<57>	48	37	14
	[25	, 57]	<48>	37	14
	[25	, 48	, 57]	<37>	14
	[25	, 37	, 48	, 57]	<14>
	[14	, 25	, 37	, 48	, 57]

算法如下：

- 1) 设置有序子表尾地址指针，待插元素指针；
- 2) 置循环计数N-1（设N个元素）；
- 3) 将待插元素插入有序子表；
- 4) 修改待插元素指针和有序表尾指针；
- 5) 计数值减1，为0吗？
- 6) 不为0则转（3）；
- 7) 为0，结束

三. 列表的种类及运算

▶ 5. 列表中的排序 — 插入排序

例子：设内存LIST开始存放一列表，表长在LLNT单元，表中元素为无符号数据，用插入排序法编程，将列表按升序排列。

1

DSEG

SEGMENT

2

LIST

DB

25,57,48,37,14

3

LLNT

DB

5

4

DSEG

ENDS

5

SSEG

SEGMENT

STACK

6

STK

DB

20 DUP (0)

7

SSEG

ENDS

8

9

CSEG

SEGMENT

10

ASSUME

CS:CSEG

11

ASSUME

DS:DSEG

12

ASSUME

SS:SSEG

13

START:

MOV

AX,DSEG

14

MOV

DS,AX

15

MOV

AX,SSEG

16

MOV

SS,AX

17

MOV

SP,SIZE STK

18

LEA

SI,LIST

19

MOV

DI,SI

20

INC

DI

21

MOV

BL,LLNT

22

DEC

BL

23

XOR

CL,CL

24

INSRT1:

INC

CL

25

PUSH

CX

26

PUSH

SI

27

PUSH

DI

28

INSRT2:

MOV

AL,[DI]

29

CMP

AL,[SI]

30

JAE

NEXT

31

MOV

AH,[SI]

32

MOV

[DI],AH

33

MOV

[SI],AL

34

DEC

SI

35

DEC

DI

36

DEC

CL

37

JNZ

INSRT2

38

NEXT:

POP

DI

39

POP

SI

40

POP

CX

41

INC

SI

42

INC

DI

43

DEC

BL

44

JNZ

INSRT1

45

MOV

AH,4CH

46

INT

21H

47

CSEG

ENDS

48

END

START

25
57
48
37
14

三. 列表的种类及运算

▶ 5. 列表中的检索操作 — 顺序检索法

例子：设内存LIST单元开始存放一列表，表中元素为字数据，表长在LLNT单元。AGE单元存放一数据。编制程序查找LIST表中是否有与该值相同的元素，若有，将其在列表中的地址存入ADR单元，否则将此单元置全1。

```
1  DSEG  SEGMENT
2  LIST  DW      35,26,7,165,47,396,47,8,105
3  LLNT  DW      8
4  AGE   DW      47
5  ADR   DW      ?
6  DSEG  ENDS
7  CSEG  SEGMENT
8  ASSUME CS:CSEG,DS:DSEG
9  START: MOV     AX,DSEG
10      MOV     DS,AX
```

```
11      LEA     DI,LIST
12      MOV     CX,LLNT
13      MOV     AX,AGE
14  SEQ2:  CMP     AX,[DI]
15      JZ      SEQ1
16      ADD     DI,2
17      LOOP    SEQ2
18      OR      DI,0FFFFH
19  SEQ1:  MOV     ADR,DI
20      MOV     AH,4CH
21      INT     21H
22  CSEG  ENDS
23      END     START
```

三. 列表的种类及运算

▶ 5. 列表中的检索操作 — 折半检索法

折半检索又称为二分检索，仅用于有序表。

例子：折半检索的过程如下图所示。图中L，R表示表的首、末地址，M表示中间位置，括号中的数字表示查找层次，查找值为57。

3	7	17	27	37	47	57	67	77	97
L(1)				M(1)					R(1)
				L(2)					
a. 中间位置M(1)=37, S>37, 查右半表									

		37	47	57	67	77	97		
		L(2)		M(2)			R(2)		
b. 中间位置M(2)=57, S=检索成功									

三. 列表的种类及运算

▶ 5. 列表中的检索操作 — 折半检索法

折半检索又称为二分检索，仅用于有序表。

例子：设内存LIST开始存放一升序表，表长在LLNT单元，表中元素为无符号字节数据，要找的元素在KEY单元。利用折半检索法，编制查找LIST表中是否有与KEY单元中内容相同的元素，若有将其在表中的地址存入ADR单元，若无，将ADR单元置全1。

```
1  DSEG  SEGMENT
2  LIST  DB      3,7,17,27,37,47,57,67,77,97
3  LLNT  DW      10
4  ADR   DW      0
5  KEY   DB      57
6  DSEG  ENDS
7  CSEG  SEGMENT
8  ASSUME CS:CSEG,DS:DSEG
9  START: MOV     AX,DSEG
10  MOV     DS,AX
```

三. 列表的种类及运算

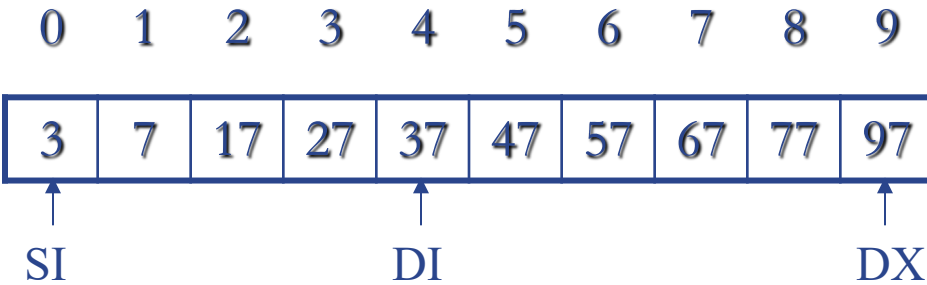
5. 列表中的检索操作 — 折半检索法

0	1	2	3	4	5	6	7	8	9
3	7	17	27	37	47	57	67	77	97
↑ SI									↑ DI

11		MOV	AL, KEY	22	BIN1:	MOV	DX, DI
12		LEA	SI, LIST	23		SUB	DI, SI
13		MOV	DI, SI	24		SHR	DI, 1
14		CMP	AL, [SI]	25		JZ	FAIL
15		JZ	SUCC	26	BIN2:	ADD	DI, SI
16		JC	FAIL	27		CMP	AL, [DI]
17		ADD	DI, LLNT	28		JZ	SUCC
18		DEC	DI	29		JC	BIN1
19		CMP	AL, [DI]	30		MOV	SI, DI
20		JZ	SUCC	31		MOV	DI, DX
21		JNC	FAIL	32		JMP	BIN1
22	BIN1:	MOV	DX, DI	33	FAIL:	MOV	DI, 0FFFFH
23		SUB	DI, SI	34	SUCC:	MOV	ADR, DI
24		SHR	DI, 1	35		MOV	AH, 4CH
25		JZ	FAIL	36		INT	21H
				37	CSEG	ENDS	
				38		END	START

三. 列表的种类及运算

▶ 5. 列表中的检索操作 — 折半检索法



```
11  MOV AL,KEY
12  LEA SI,LIST
13  MOV DI,SI
14  CMP AL,[SI]
15  JZ  SUCC
16  JC  FAIL
17  ADD DI,LLNT
18  DEC DI
19  CMP AL,[DI]
20  JZ  SUCC
21  JNC FAIL
22  BIN1: MOV DX,DI
23       SUB DI,SI
24       SHR DI,1
25       JZ  FAIL
26  BIN2: ADD DI,SI
27       CMP AL,[DI]
28       JZ  SUCC
29       JC  BIN1
30       MOV SI,DI
31       MOV DI,DX
32       JMP BIN1
33  FAIL: MOV DI,0FFFFH
34  SUCC: MOV ADR,DI
35       MOV AH,4CH
36       INT 21H
37  CSEG  ENDS
38       END  START
```

三. 列表的种类及运算

▶ 5. 列表中的检索操作 — 折半检索法

0	1	2	3	4	5	6	7	8	9
3	7	17	27	37	47	57	67	77	97
				↑ SI					↑ DI

11		MOV	AL,KEY	22	BIN1:	MOV	DX,DI
12		LEA	SI,LIST	23		SUB	DI,SI
13		MOV	DI,SI	24		SHR	DI,1
14		CMP	AL,[SI]	25		JZ	FAIL
15		JZ	SUCC	26	BIN2:	ADD	DI,SI
16		JC	FAIL	27		CMP	AL,[DI]
17		ADD	DI,LLNT	28		JZ	SUCC
18		DEC	DI	29		JC	BIN1
19		CMP	AL,[DI]	30		MOV	SI,DI
20		JZ	SUCC	31		MOV	DI,DX
21		JNC	FAIL	32		JMP	BIN1
22	BIN1:	MOV	DX,DI	33	FAIL:	MOV	DI,0FFFFH
23		SUB	DI,SI	34	SUCC:	MOV	ADR,DI
24		SHR	DI,1	35		MOV	AH,4CH
25		JZ	FAIL	36		INT	21H
				37	CSEG	ENDS	
				38		END	START

三. 列表的种类及运算

▶ 5. 列表中的检索操作 — 折半检索法

0	1	2	3	4	5	6	7	8	9
3	7	17	27	37	47	57	67	77	97
				↑		↑			↑
				SI		DI			DX

```
11      MOV     AL,KEY
12      LEA     SI,LIST
13      MOV     DI,SI
14      CMP     AL,[SI]
15      JZ      SUCC
16      JC      FAIL
17      ADD     DI,LLNT
18      DEC     DI
19      CMP     AL,[DI]
20      JZ      SUCC
21      JNC     FAIL
22  BIN1:  MOV     DX,DI
23      SUB     DI,SI
24      SHR     DI,1
25      JZ      FAIL
```

```
22  BIN1:  MOV     DX,DI
23      SUB     DI,SI
24      SHR     DI,1
25      JZ      FAIL
26  BIN2:  ADD     DI,SI
27      CMP     AL,[DI]
28      JZ      SUCC
29      JC      BIN1
30      MOV     SI,DI
31      MOV     DI,DX
32      JMP     BIN1
33  FAIL:  MOV     DI,0FFFFH
34  SUCC:  MOV     ADR,DI
35      MOV     AH,4CH
36      INT     21H
37  CSEG   ENDS
38      END     START
```

四. 字符串处理

▶ 1. 字符串的定义及表示方法

字符串是一种特殊的线性表，它的数据元素仅由一个字符组成。

一般地讲，字符串是零个或多个字符的有限序列。一个字符串中包含的字符个数，称为这个字符串的长度。长度为零的字符串称为空串，它不包含任何字符。通常用单引号或双引号将字符串括起来，例如：“X1”，‘ABC’

在8086/8088宏汇编语言中，字符串的表示可以借助数据定义语句DB来定义，字符串的特征表示一般有两种，一种是在字符串的尾部用00H作标记，另一种是在字符串的前面空出一个单元，放置字符串的长度。例：

```
STR1  DB    'I am a student', 0
```

```
STR2  DB    4, 'ABCD'
```


四. 字符串处理

▶ 2. 字符串的运算

对字符串的基本运算有：

- 1) 求字符串的长度；
- 2) 字符串的比较；
- 3) 字符串的连接；
- 4) 取子字符串；
- 5) 取子字符串序号。

除以上基本运算外，还有以下基本运算的组合运算：

- 1) 插入；
- 2) 删除；
- 3) 查找并替换。

四. 字符串处理

▶ 2. 字符串的运算

例子：内存STR单元存放一字符串，结尾用00H作标记。编程求其串长并存入LEN单元。

```
1  DSEG  SEGMENT
2  STR   DB  'I live in Shenyang',00H
3  LEN   DB  ?
4  DSEG  ENDS
5  CSEG  SEGMENT
6  ASSUME CS:CSEG,DS:DSEG
7  SLEN:  MOV  AX,DSEG
8         MOV  DS,AX
9         LEA  SI,STR
10        XOR  AL,AL
11  LOP:  CMP  BYTE PTR[SI],00H
12        JZ   DONE
13        INC  AL
14        INC  SI
15        JMP  LOP
16  DONE: MOV  LEN,AL
17        MOV  AH,4CH
18        INT  21H
19  CSEG  ENDS
20        END  SLEN
```

四. 字符串处理

3. 字符串的比较

例子：内存STR1和STR2单元分别存放两个字符串，均以00H作结束标记。编制程序比较两个字符串，比较结果存于FLAG单元。如果两串相等，FLAG为0；如果STR1>STR2，FLAG为1；如果STR1<STR2，FLAG为FFH。

```
1  DSEG      SEGMENT
2  STR1      DB  'I am a student',00H
3  STR2      DB  'I AM A STUDENT',00H
4  FLAG      DB  ?
5  DSEG      ENDS
6  CSEG      SEGMENT
7             ASSUME  CS:CSEG, DS:DSEG
8  BSTR:     MOV     AX, DSEG
9             MOV     DS, AX
10            LEA     SI, STR1
11            LEA     BX, STR2
12            MOV     FLAG, 0
13            MOV     CL, 0
14  LOP:     XOR     AX, AX
15            CMP     [SI], AL
16            JZ      DONE1
17            MOV     AH, 1
18  DONE1:   CMP     [BX], CL
19            JZ      DONE2
20            MOV     AL, 1
21  DONE2:   OR      AH, AL
22            JZ      DONE
23            MOV     CH, [SI]
24            CMP     CH, [BX]
25            JNZ     DONE3
26            INC     BX
27            INC     SI
28            JMP     LOP
29  DONE3:   JA      DONE4
30            DEC     FLAG
31            JMP     DONE
32  DONE4:   INC     FLAG
33  DONE:    MOV     AH, 4CH
34            INT     21H
35  CSEG      ENDS
36            END     BSTR
```

课后作业



1. 用记录定义下述表格，要求
- 代码范围：1-100
 - 学位：学士、硕士、博士
 - 政治面貌：团员、党员、预备党员、群众
 - 自己设计20个记录，将其中的20年以上教龄、男性且具有博士学位人员的记录选取并存入PERS开始的存储区。

工号	性别	学位	民族	政治面貌	是否20年以上教龄

课后作业



2. 用结构定义下述表格，要求

- 器材编号：1-1000
- 生产工厂：20个字符
- 单价范围：1-100
- 数量范围：0-1000
- 购入年份：2000-2020
- 自己定义20种器材数据，计算单价为50元以上、总金额超过1000元以上的器材总金额。

器材编号	成产工厂	单价	数量	购入年份	总金额

课后作业



3. 编写程序，将两个长度不同的字符串STR1和STR2在屏幕上以左边对齐的方式显示出来，设各字符串长度小于40字符。