

1. 除法输出（二进制）

```
MOV AX,1234H ;把要输出的数据放入 AX  
MOV CX,16 ;循环次数，输出几位循环几次  
MOV DX,0 ;使用除法 DIV,高位放在 DX,低位放在 AX  
MOV BX,1000000000000000B ;除法的基数  
AGAIN:  
DIV BX ;循环标志，从以下开始循环  
MOV SI,DX ;除以基数，获得某位，为输出做准备  
MOV DL,AL ;DIV 余数放在 DX，要保护好  
ADD DL,30H ;要输出的位从 AL 挪到 DL  
MOV AH,2 ;转换 ASCII 码，加 30H(十六进制再加 7H)  
INT 21H ;调用输出程序  
MOV AX,BX ;对基数进行更新  
MOV DX,0  
MOV BX,2  
DIV BX  
MOV BX,AX ;更新后的基数再次放入  
MOV AX,SI ;上一次输出后的余数放入 AX，循环  
LOOP AGAIN
```

2. 除法输出（十进制）

```
MOV AX,12345  
MOV CX,5  
MOV DX,0  
MOV BX,10000  
AGAIN:  
DIV BX  
MOV SI,DX  
MOV DL,AL  
ADD DL,30H  
MOV AH,2  
INT 21H  
MOV AX,BX  
MOV DX,0  
MOV BX,10  
DIV BX  
MOV BX,AX  
MOV AX,SI  
LOOP AGAIN
```

3. 除法输出（十六进制）

```
MOV AX,0ABCDH  
MOV CX,4  
MOV DX,0
```

```

MOV BX,1000H
AGAIN:
DIV BX
MOV SI,DX
MOV DL,AL
ADD DL,30H
CMP DL,3AH ;DL 中的数与 3AH 做比较， 小于不加 7
JB NOADD7
ADD DL,7H
NOADD7:
MOV AH,2
INT 21H
MOV AX,BX
MOV DX,0
MOV BX,16
DIV BX
MOV BX,AX
MOV AX,SI
LOOP AGAIN

```

4. 循环输出（十六进制）

```

MOV BX,12ABH ;把要输出的数字放入 BX
MOV SI,4 ;循环输出时要用 CL 存放循环位数
NEXT:
MOV CL,4
ROL BX,CL ;循环移位
MOV AL,BL
AND AL,00001111B ;把最高位移到最低位， 与 0FH 求与
ADD AL,30H
CMP AL,3AH
JB NOADD7
ADD AL,7
NOADD7:
MOV DL,AL
MOV AH,2
INT 21H
DEC SI ;SI 自减一， 判断是否为 0 终止循环
JNZ NEXT

```

5. 循环输出（二进制）

```

MOV BX,12ABH
MOV CX,16 ;每次循环一位， 不需要用 CX
NEXT:
ROL BX,1

```

```
MOV AL,BL  
AND AL,00000001B  
ADD AL,30H  
MOV DL,AL  
MOV AH,2  
INT 21H  
LOOP NEXT
```

6. 乘积输入（输入十六，转十进制）

```
MOV BX,0  
NEXT:  
MOV AH,1  
INT 21H  
CMP AL,0DH ;与回车比较，输入回车跳转退出  
JZ ENDIN  
SUB AL,30H  
CMP AL,0AH ;ASCII 码转数字  
JB NOSUB7  
SUB AL,7  
NOSUB7:  
MOV AH,0  
MOV SI,10  
XCHG AX,BX ;BX=10*BX+AX  
MUL SI  
XCHG AX,BX  
ADD BX,AX  
JMP NEXT  
ENDIN:
```

7. 乘积输入（输入十，转十进制）

```
MOV BX,0  
NEXT:  
MOV AH,1  
INT 21H  
CMP AL,0DH  
JZ ENDIN  
SUB AL,30H  
MOV AH,0  
MOV SI,10  
XCHG AX,BX  
MUL SI  
XCHG AX,BX  
ADD BX,AX  
JMP NEXT
```

ENDIN:

8. 乘积输入（输入十六，转十六进制）

MOV BX,0

NEXT:

MOV AH,1

INT 21H

CMP AL,0DH

JZ ENDIN

SUB AL,30H

CMP AL,0AH

JB NOSUB7

SUB AL,7

NOSUB7:

MOV AH,0

MOV SI,16

XCHG AX,BX

MUL SI

XCHG AX,BX

ADD BX,AX

JMP NEXT

ENDIN:

9. 计算 W=X+Y-Z (均为 32 位数, 高精度计算以及强制类型转换)

MOV AX,WORD PTR[X]

ADD AX,WORD PTR[Y]

MOV DX,WORD PTR[X+2]

ADC AX,WORD PTR[Y+2]

ADD AX,24

ADC DX,0

SUB AX,WORD PTR[Z]

SBB DX,WORD PTR[Z+2]

MOV WORD PTR[W],AX

MOV WORD PTR[W+2],DX

10.串的复制 (均为 8 位)

LEA SI,ARR1

;也可用 MOV SI,OFFSET ARR1

LEA DI,ARR2

MOV CX,4

CLD

AGAIN:

MOVSB

LOOP AGAIN

11. 汇编程序框架

```
DATAS SEGMENT  
    ;此处输入数据段代码  
DATAS ENDS  
  
STACKS SEGMENT  
    ;此处输入堆栈段代码  
STACKS ENDS  
  
CODES SEGMENT  
    ASSUME CS:CODES,DS:DATAS,SS:STACKS  
START:  
    MOV AX,DATAS  
    MOV DS,AX  
    ;此处输入代码段代码  
    MOV AH,4CH  
    INT 21H  
CODES ENDS  
END START
```

12. 输入十六进制带符号温度值，存储小于 60 的个数并以十进制输出

```
DATAS SEGMENT  
    ;此处输入数据段代码  
    CNT DW 0  
DATAS ENDS  
  
STACKS SEGMENT  
    ;此处输入堆栈段代码  
STACKS ENDS  
  
CODES SEGMENT  
    ASSUME CS:CODES,DS:DATAS,SS:STACKS  
INPUT PROC  
  
    MOV BX,0  
    MOV CX,4  
NEXT:  
    MOV AH,1  
    INT 21H  
    SUB AL,30H  
    CMP AL,0AH  
    JB NOSUB7  
    SUB AL,7
```

```
NOSUB7:  
    MOV AH,0  
    MOV SI,16  
    XCHG AX,BX  
    MUL SI  
    XCHG AX,BX  
    ADD BX,AX  
    LOOP NEXT  
    RET  
    INPUT ENDP  
  
START:  
    MOV AX,DATAS  
    MOV DS,AX  
    ;输入子程序,结果放入 BX  
    MOV DI,5  
    COUNT:  
    CALL INPUT  
    CMP BX,60  
    JG NOCNT  
    INC CNT  
    NOCNT:  
    DEC DI  
    JNZ COUNT  
    ;输出结果,结果从 CNT 放入 AX  
    MOV AX,CNT  
    MOV CX,5  
    MOV DX,0  
    MOV BX,10000  
    COUT:  
    DIV BX  
    MOV SI,DX  
    MOV DL,AL  
    ADD DL,30H  
    MOV AH,2  
    INT 21H  
    MOV AX,BX  
    MOV DX,0  
    MOV BX,10  
    DIV BX  
    MOV BX,AX  
    MOV AX,SI  
    LOOP COUT  
  
    MOV AH,4CH
```

```
INT 21H  
CODES ENDS  
END START
```

13.将 ARRAY 数组中的值求平均值的 150%并比较，存储并输出小于基值正常值和总个数

```
DATAS SEGMENT  
;此处输入数据段代码  
ARRAY DW  
20,1266,1276,2802,1260,1322,1289,1333,3056,1311,1321,1255,1231,1232,1275  
DW 1286,1299,3356,1301,1289,1302  
NORMAL DW ?  
DATAS ENDS
```

```
STACKS SEGMENT  
;此处输入堆栈段代码  
STACKS ENDS
```

```
CODES SEGMENT  
ASSUME CS:CODES,DS:DATAS,SS:STACKS  
START:  
MOV AX,DATAS  
MOV DS,AX  
;此处输入代码段代码  
MOV CX,21  
MOV DX,0  
MOV SI,0  
CNT:  
ADD DX,ARRAY[SI]  
ADD SI,2  
LOOP CNT  
MOV AX,DX  
MOV DX,0  
MOV BX,21  
DIV BX  
MOV DX,0  
MOV BX,2  
DIV BX  
MOV BX,3  
MUL BX  
;求得平均值的百分之 50，并放入 AX  
MOV CX,21  
MOV DX,0;存放正常数据的数量  
MOV DI,2;偏移地址
```

```
MOV SI,0
COUNT:
MOV BX,ARRAY[SI]
CMP BX,AX
JA NOCNT;小于就计数一次， normal 指针移动
INC DX
MOV NORMAL[DI],BX
ADD DI,2
NOCNT:
ADD SI,2;不管结果如何， si 一定要加 1
LOOP COUNT
MOV NORMAL,DX;把计数结果加入 normal 的首地址
;输出
MOV SP,DX
ADD SP,1
MOV DI,0
OUTPUT:
MOV AX,NORMAL[DI]
;输出程序
MOV CX,5
MOV DX,0
MOV BX,10000
COUT:
DIV BX
MOV SI,DX
MOV DL,AL
ADD DL,30H
MOV AH,2
INT 21H
MOV AX,BX
MOV DX,0
MOV BX,10
DIV BX
MOV BX,AX
MOV AX,SI
LOOP COUT
;偏移地址自动增加， 大循环次数减少
ADD DI,2
SUB SP,1
;输出用回车分割
MOV DL,''
MOV AH,2
INT 21H
;BP 不是 0 持续输出
```

```
CMP SP,0  
JA OUTPUT  
MOV AH,4CH  
INT 21H  
CODES ENDS  
END START
```

14. 输入一个年份，并判断是否是闰年并输出结果

```
DATAS SEGMENT  
;此处输入数据段代码  
RU DB 'I','R','U','N','N','I','A','N'  
NR DB 'I','N','O','I','R','U','N','N','I','A','N'  
DATAS ENDS
```

```
STACKS SEGMENT  
;此处输入堆栈段代码  
STACKS ENDS
```

```
CODES SEGMENT  
ASSUME CS:CODES,DS:DATAS,SS:STACKS  
START:  
    MOV AX,DATAS  
    MOV DS,AX  
;此处输入代码段代码  
;输入一个年份 4 位十进制  
    MOV BX,0  
    MOV CX,4  
NEXT:  
    MOV AH,1  
    INT 21H  
    SUB AL,30H  
    MOV AH,0  
    MOV SI,10  
    XCHG AX,BX  
    MUL SI  
    XCHG AX,BX  
    ADD BX,AX  
LOOP NEXT  
    MOV DI,BX  
    MOV DL,' '  
    MOV AH,2  
    INT 21H  
;除法输出年份  
    MOV AX,BX
```

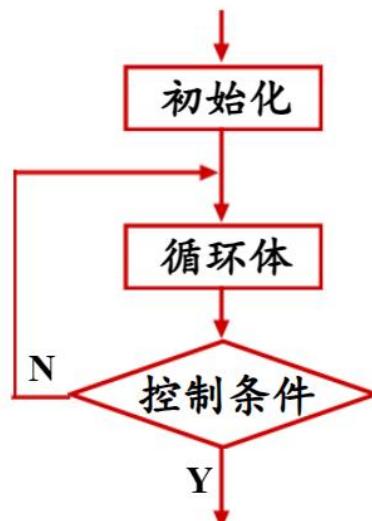
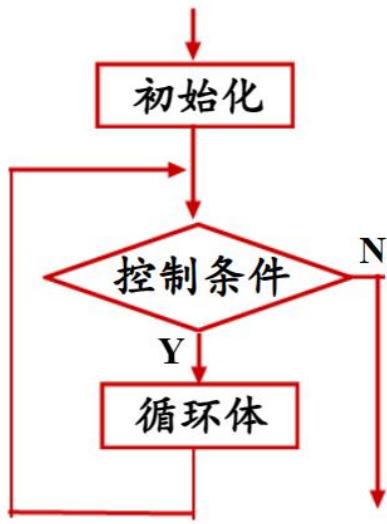
```
MOV CX,4
MOV DX,0
MOV BX,1000
AGAIN:
DIV BX
MOV SI,DX
MOV DL,AL
ADD DL,30H
MOV AH,2
INT 21H
MOV AX,BX
MOV DX,0
MOV BX,10
DIV BX
MOV BX,AX
MOV AX,SI
LOOP AGAIN
MOV SI,DI
;判断是否是闰年并输出
MOV DX,0
MOV AX,SI
MOV BX,100
DIV BX
CMP DX,0
JE SECOND
MOV DX,0
MOV AX,SI
MOV BX,4
DIV BX
CMP DX,0
JE RUNNIAN
;输出不是闰年
JMP NORUNNIAN
SECOND:
MOV DX,0
MOV AX,SI
MOV BX,400
DIV BX
CMP DX,0
JE RUNNIAN
;输出不是闰年
JMP NORUNNIAN
;闰年输出
RUNNIAN:
```

```

MOV DI,0
MOV CX,8
ONEMORE:
MOV DL,RU[DI]
MOV AH,2
INT 21H
INC DI
LOOP ONEMORE
JMP EXIT
;不是闰年输出
NORUNNIAN:
MOV DI,0
MOV CX,11
ONCEMORE:
MOV DL,NR[DI]
MOV AH,2
INT 21H
INC DI
LOOP ONCEMORE
EXIT:
MOV AH,4CH
INT 21H
CODES ENDS
END START

```

15. 流程图模板



DO-WHILE 结构

DO-UNTIL 结构

16.ASCII 码表（可显示字符）

ASCII可显示字符（共95个）

二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形
0010 0000	32	20	(space)	0100 0000	64	40	@	0110 0000	96	60	'
0010 0001	33	21	!	0100 0001	65	41	A	0110 0001	97	61	a
0010 0010	34	22	"	0100 0010	66	42	B	0110 0010	98	62	b
0010 0011	35	23	#	0100 0011	67	43	C	0110 0011	99	63	c
0010 0100	36	24	\$	0100 0100	68	44	D	0110 0100	100	64	d
0010 0101	37	25	%	0100 0101	69	45	E	0110 0101	101	65	e
0010 0110	38	26	&	0100 0110	70	46	F	0110 0110	102	66	f
0010 0111	39	27	'	0100 0111	71	47	G	0110 0111	103	67	g
0010 1000	40	28	(0100 1000	72	48	H	0110 1000	104	68	h
0010 1001	41	29)	0100 1001	73	49	I	0110 1001	105	69	i
0010 1010	42	2A	*	0100 1010	74	4A	J	0110 1010	106	6A	j
0010 1011	43	2B	+	0100 1011	75	4B	K	0110 1011	107	6B	k
0010 1100	44	2C	,	0100 1100	76	4C	L	0110 1100	108	6C	l
0010 1101	45	2D	-	0100 1101	77	4D	M	0110 1101	109	6D	m
0010 1110	46	2E	.	0100 1110	78	4E	N	0110 1110	110	6E	n
0010 1111	47	2F	/	0100 1111	79	4F	O	0110 1111	111	6F	o
0011 0000	48	30	0	0101 0000	80	50	P	0111 0000	112	70	p
0011 0001	49	31	1	0101 0001	81	51	Q	0111 0001	113	71	q
0011 0010	50	32	2	0101 0010	82	52	R	0111 0010	114	72	r
0011 0011	51	33	3	0101 0011	83	53	S	0111 0011	115	73	s
0011 0100	52	34	4	0101 0100	84	54	T	0111 0100	116	74	t
0011 0101	53	35	5	0101 0101	85	55	U	0111 0101	117	75	u
0011 0110	54	36	6	0101 0110	86	56	V	0111 0110	118	76	v
0011 0111	55	37	7	0101 0111	87	57	W	0111 0111	119	77	w
0011 1000	56	38	8	0101 1000	88	58	X	0111 1000	120	78	x
0011 1001	57	39	9	0101 1001	89	59	Y	0111 1001	121	79	y
0011 1010	58	3A	:	0101 1010	90	5A	Z	0111 1010	122	7A	z
0011 1011	59	3B	;	0101 1011	91	5B	[0111 1011	123	7B	{
0011 1100	60	3C	<	0101 1100	92	5C	\	0111 1100	124	7C	
0011 1101	61	3D	=	0101 1101	93	5D]	0111 1101	125	7D	}
0011 1110	62	3E	>	0101 1110	94	5E	^	0111 1110	126	7E	~
0011 1111	63	3F	?	0101 1111	95	5F	_				

17.ASCII 码表（控制字符）

ASCII控制字符 (共33个)

二进制	十进制	十六进制	缩写	Unicode 表示法	脱出字符 表示法	名称 / 意义
0000 0000	0	00	NUL	„NUL„	^@	空字符 (Null)
0000 0001	1	01	SOH	„SOH„	^A	标题开始
0000 0010	2	02	STX	„STX„	^B	本文开始
0000 0011	3	03	ETX	„ETX„	^C	本文结束
0000 0100	4	04	EOT	„EOT„	^D	传输结束
0000 0101	5	05	ENQ	„ENQ„	^E	请求
0000 0110	6	06	ACK	„ACK„	^F	确认回应
0000 0111	7	07	BEL	„BEL„	^G	响铃
0000 1000	8	08	BS	„BS„	^H	退格
0000 1001	9	09	HT	„HT„	^I	水平定位符号
0000 1010	10	0A	LF	„LF„	^J	换行键
0000 1011	11	0B	VT	„VT„	^K	垂直定位符号
0000 1100	12	0C	FF	„FF„	^L	换页键
0000 1101	13	0D	CR	„CR„	^M	CR (字符)
0000 1110	14	0E	SO	„SO„	^N	取消变换 (Shift out)
0000 1111	15	0F	SI	„SI„	^O	启用变换 (Shift in)
0001 0000	16	10	DLE	„DLE„	^P	跳出数据通讯
0001 0001	17	11	DC1	„DC1„	^Q	设备控制一 (XON 激活软件速度控制)
0001 0010	18	12	DC2	„DC2„	^R	设备控制二
0001 0011	19	13	DC3	„DC3„	^S	设备控制三 (XOFF 停用软件速度控制)
0001 0100	20	14	DC4	„DC4„	^T	设备控制四
0001 0101	21	15	NAK	„NAK„	^U	确认失败回应
0001 0110	22	16	SYN	„SYN„	^V	同步用暂停
0001 0111	23	17	ETB	„ETB„	^W	区块传输结束
0001 1000	24	18	CAN	„CAN„	^X	取消
0001 1001	25	19	EM	„EM„	^Y	连线介质中断
0001 1010	26	1A	SUB	„SUB„	^Z	替换
0001 1011	27	1B	ESC	„ESC„	^_	退出键
0001 1100	28	1C	FS	„FS„	^`	文件分割符
0001 1101	29	1D	GS	„GS„	^]	组群分隔符
0001 1110	30	1E	RS	„RS„	^^	记录分隔符
0001 1111	31	1F	US	„US„	^-	单元分隔符
0111 1111	127	7F	DEL	„DEL„	^?	Delete字符