**《汇编上机题源码》**

**第一套**

DATAS SEGMENT

;此处输入数据段代码

INPUT\_REMIND DB 'Please input a year ','$' ;输入提示

INPUT\_ERROR DB 'Input error',10,'$' ;边界判断

IS\_RUN DB 'The year is run year',10,'$' ;是闰年

NOT\_RUN DB 'The year is not run year',10,'$' ;不是闰年

BUF DB 10 ;数据缓冲区

DB ?

DB 10 DUP(?)

LONG EQU [BUF+1] ;字符串真正长度

NUM DW ? ;辅助数字

QIAN DW 1000 ;千

BAI DB 100 ;百

SHI DB 10 ;十

GEWEI DB 1 ;个

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 DX,OFFSET INPUT\_REMIND ;输入提示

MOV AH,09H

INT 21H

MOV DX,OFFSET BUF ;输入数据

MOV AH,0AH

INT 21H

MOV AL,LONG

CMP AL,4

JZ TRUE ;长度不等于4的数据都有误

ERROR: ;输入数据有误

MOV DL,10 ;换行

MOV AH,02H

INT 21H

MOV DX,OFFSET INPUT\_ERROR ;输出错误提示

MOV AH,09H

INT 21H

JMP EXIT

TRUE: ;输入数据正确时

MOV AX,WORD PTR [BUF+2]

SUB AX,30H

MUL QIAN

MOV NUM,AX

MOV AL,[BUF+3]

SUB AL,30H

MUL BAI

ADD NUM,AX

MOV AL,[BUF+4]

SUB AL,30H

MUL SHI

ADD NUM,AX

MOV AL,[BUF+5]

SUB AL,30H

MUL GEWEI

ADD NUM,AX ;此处年份已经被转换为数字

MOV BX,400

MOV DX,0

MOV AX,NUM

DIV BX ;商存放在AX中，余数存放在DX中

CMP DX,0

JZ RUN\_YEAR

JMP JUDGE

JUDGE:

MOV BX,4

MOV DX,0

MOV AX,NUM

DIV BX

CMP DX,0

JZ JUDGE2

JMP NOT\_YEAR

JUDGE2:

MOV BX,100

MOV DX,0

MOV AX,NUM

DIV BX

CMP DX,0

JZ NOT\_YEAR

JMP RUN\_YEAR

RUN\_YEAR: ;是闰年

MOV DL,10

MOV AH,02H

INT 21H

MOV DX,OFFSET IS\_RUN

MOV AH,09H

INT 21H

JMP EXIT

NOT\_YEAR: ;不是闰年

MOV DL,10

MOV AH,02H

INT 21H

MOV DX,OFFSET NOT\_RUN

MOV AH,09H

INT 21H

JMP EXIT

EXIT: ;退出

MOV AH,4CH

INT 21H

CODES ENDS

END START

**第二套**

DATAS SEGMENT

;此处输入数据段代码

REMIND DB 'Enter to input wanshu',10,'$'

SUM DW ? ;所有因子的和

X DW ? ;要判断的数

I DW ? ;判断因子的除数

REMINDER DB ? ;余数

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 DX,OFFSET REMIND

MOV AH,09H

INT 21H

MOV AH,07H

INT 21H

CMP AL,13 ;判断是否为回车键

MOV SUM,0 ;初始化因子和

MOV X,1 ;初始化第一个数

JZ TRUE

JMP EXIT

TRUE:

MOV I,1 ;除数初始化为1

JMP JUDGE

X\_LOOP:

INC X

MOV SUM,0 ;先清零,再判断下一个数

MOV AX,X

CMP AX,10000 ;控制数的范围

JBE TRUE

JMP EXIT

JUDGE:

MOV BX,I

MOV DX,0

MOV AX,X ;被除数为32位的

DIV BX

CMP DX,0 ;判断余数是否为0

JZ SUM\_ADD

JMP DIV\_I

SUM\_ADD: ;求各个因子的和

MOV AX,I

ADD SUM,AX

JMP DIV\_I

DIV\_I: ;该循环负责控制求因子的循环

INC I

MOV BX,I

CMP BX,X

JB JUDGE

MOV BX,SUM

CMP BX,X

JZ IS\_WANSHU

JMP X\_LOOP

IS\_WANSHU:

MOV DL,10

MOV AH,02H

INT 21H

MOV DX,X

CMP DX,10

JB OUT\_10

JA OUT\_100

OUT\_10:

ADD DX,30H

MOV AH,02H

INT 21H

JMP X\_LOOP

OUT\_100:

MOV DX,X

CMP DX,1000

JA EXIT

MOV AX,DX

MOV BL,100

DIV BL

MOV DL,AL ;输出百位

MOV REMINDER,AH ;余数先保护起来

ADD DL,30H

MOV AH,02H

INT 21H

MOV AX,WORD PTR REMINDER

MOV BL,10

DIV BL

MOV DL,AL

MOV REMINDER,AH

ADD DL,30H

MOV AH,02H

INT 21H

MOV DL,REMINDER

ADD DL,30H

MOV AH,02H

INT 21H

JMP X\_LOOP

EXIT:

MOV AH,4CH

INT 21H

CODES ENDS

END START

**第三套**

DATAS SEGMENT

;此处输入数据段代码

REMIND DB 'Please input a string and the length is between 15~80',10,'$';用户输入提示

ERROR\_INPUT DB 'input error!',10,'$' ;用户输入字符串长度不对

NOT\_CHAR\_NUM DB 'the number of not char is ','$' ;非字母的字符个数

MAX\_ASCII\_OUT DB 'the max ascii of this string is ','$';最大的ASCII码

BUF DB 100 ;输入缓冲区

DB ?

DB 100 DUP(?)

LONG EQU BUF+1 ;等值语句，LONG为字符串实际长度

SUM DW ? ;字符串中非字母的个数

COUNT DB ? ;计数器

MAX DB ? ;最大的ASCII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 DX,OFFSET REMIND ;输入信息提示

MOV AH,09H

INT 21H

MOV DX,OFFSET BUF ;字符串缓冲

MOV AH,0AH

INT 21H

MOV DL,10 ;换行

MOV AH,02H

INT 21H

CMP LONG,15 ;串长小于15

JL ERROR

CMP LONG,80 ;串长大于80

JG ERROR

MOV COUNT,0 ;计数器COUNT清零

MOV SUM,0 ;非字母的个数初始化

MOV SI,1 ;字符串指针初始化

MOV MAX,0 ;假定0是ASCII码最大的

JMP ASSCII

ERROR: ;错误的输入提示

MOV DX,OFFSET ERROR\_INPUT

MOV AH,09H

INT 21H

JMP OVER

ASSCII:

MOV DX,0 ;找ASCII码最大值

MOV DL,MAX

CMP [BUF+SI],DL

JA CHANGE\_MAX ;当前数大于MAX,修改

JB TRUE ;当前数小于MAX，不修改

CHANGE\_MAX:

MOV DX,0

MOV DL,[BUF+SI]

MOV MAX,DL ;修改最大值MAX

JMP TRUE

TRUE: ;正确的输入情况下

INC SI

CMP [BUF+SI],65 ;A以下的字符

JB NOT\_CHAR

CMP [BUF+SI],122 ;z以上的字符

JA NOT\_CHAR

CMP [BUF+SI],90 ;Z以上的字符

JA MAY\_NOT\_CHAR

MOV DL,10

MOV AH,02H

INT 21H

MOV DX,0 ;清零

MOV DL,[BUF+SI] ;字母正常显示

MOV AH,02H

INT 21H

JMP JUDGE

NOT\_CHAR: ;非字母

INC SUM ;非字母的个数加1

MOV DL,10

MOV AH,02H

INT 21H

MOV AX,0 ;清零

MOV AH,09H ;让非字母的闪烁显示

MOV AL,[BUF+SI]

MOV BX,0 ;清零

MOV BH,0

MOV BL,10000010B ;背景为黑，前景为绿，闪烁显示

MOV CX,1

INT 10H

JMP JUDGE

MAY\_NOT\_CHAR: ;可能不是字母

CMP [BUF+SI],97 ;a以下的字符

JB NOT\_CHAR

MOV DL,10

MOV AH,02H

INT 21H

MOV DX,0 ;清零

MOV DL,[BUF+SI] ;字母正常显示

MOV AH,02H

INT 21H

JMP JUDGE

JUDGE: ;判断循环结束

INC COUNT

MOV CX,0 ;首先清空CX寄存器

MOV CL,COUNT

CMP LONG,CL ;字符串长度和计数器比较

JG ASSCII ;循环

JMP SHOW\_SUM

SHOW\_SUM: ;输出不是字母的个数

MOV DL,10 ;换行

MOV AH,02H

INT 21H

MOV DX,OFFSET NOT\_CHAR\_NUM ;输出提示

MOV AH,09H

INT 21H

MOV DX,0

MOV DX,SUM

ADD DX,30H

MOV AH,02H

INT 21H

MOV DL,10 ;换行

MOV AH,02H

INT 21H

MOV DX,OFFSET MAX\_ASCII\_OUT

MOV AH,09H

INT 21H

MOV AX,0 ;清零

MOV AH,09H ;让非字母的闪烁显示

MOV AL,MAX

MOV BX,0 ;清零

MOV BH,0

MOV BL,00000100B ;背景为黑，前景为红，不闪烁显示

MOV CX,1

INT 10H

OVER:

MOV AH,4CH

INT 21H

CODES ENDS

END START

**第四套**

DATAS SEGMENT

;此处输入数据段代码

INPUT\_REMIND DB 'please input a string above 15',10,'$' ;输入提示语句

ERROR\_INPUT DB 'input error',10,'$' ;错误输入

SEARCH\_INPUT DB 'please input what string you want to search',10,'$' ;输入检索字符串

OUT\_NUM DB 'the num of your string is ','$' ;检索字符串的和

BUF1 DB 100 ;输入字符串缓存区

DB ?

DB 100 DUP(?)

LONG1 EQU [BUF1+1] ;输入字符串的真正长度

COUNT1 DB ? ;计数器1

BUF2 DB 100 ;待检索字符串缓存区

DB ?

DB 100 DUP(?)

LONG2 EQU [BUF2+1] ;待检索检索字符串的真正长度

COUNT2 DB ? ;计数器2

NUM DB ? ;待检索字符串出现的次数

COUNT3 DB ? ;计数器3

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 DX,OFFSET INPUT\_REMIND ;输入提示

MOV AH,09H

INT 21H

MOV DX,OFFSET BUF1 ;输入缓存

MOV AH,0AH

INT 21H

CMP LONG1,5 ;判断所输入的字符串长度是否正确

JB ERROR

JMP SEARCH

ERROR: ;报错

MOV DX,0

MOV DL,10 ;换行

MOV AH,02H

INT 21H

MOV DX,OFFSET ERROR\_INPUT ;输出错误提示信息

MOV AH,09H

INT 21H

JMP EXIT ;结束

SEARCH: ;提示输入检索字符串

MOV DX,0

MOV DL,10 ;换行

MOV AH,02H

INT 21H

MOV DX,OFFSET SEARCH\_INPUT ;输入提示

MOV AH,09H

INT 21H

MOV DX,OFFSET BUF2 ;字符缓存

MOV AH,0AH

INT 21H

MOV DL,10 ;换行

MOV AH,02H

INT 21H

MOV COUNT1,0 ;计数器初始化

MOV SI,2 ;SI指针负责输入字符串的访问

MOV COUNT2,0

MOV DI,2 ;DI指针负责要查找字符串的访问

MOV COUNT3,0

TRUE:

MOV AL,[BUF2+DI]

CMP AL,[BUF1+SI]

JZ FIRST\_CHAR

JMP MIDDLE

MIDDLE:

CMP DI,2

JZ LONG

POP SI

MOV DI,2

MOV COUNT3,0

JMP LONG

LONG:

MOV DL,[BUF1+SI]

MOV AH,02H

INT 21H

CALL CHANGE\_LINE

INC SI

INC COUNT1

MOV AL,COUNT1

CMP AL,LONG1

JB TRUE

JMP SHOW

FIRST\_CHAR:

CMP DI,2

JZ TAG

JMP SHORT\_STRING

TAG:

PUSH SI

JMP SHORT\_STRING

SHORT\_STRING:

INC SI

INC COUNT1

INC DI

INC COUNT2

MOV AL,COUNT2

CMP AL,LONG2

JB TRUE

POP SI

JMP COLOR\_OUTPUT

COLOR\_OUTPUT:

MOV AH,09H

MOV BH,0

MOV BL,10000100B

MOV AL,[BUF1+SI]

MOV CX,1

INT 10H

CALL CHANGE\_LINE

INC SI

INC COUNT3

MOV AL,COUNT3

CMP AL,LONG2

JB COLOR\_OUTPUT

MOV DI,2

MOV COUNT3,0

INC NUM

JMP TRUE

SHOW:

MOV DX,OFFSET OUT\_NUM

MOV AH,09H

INT 21H

MOV DL,NUM

ADD DL,30H

MOV AH,02H

INT 21H

JMP EXIT

EXIT:

MOV AH,4CH

INT 21H

CHANGE\_LINE PROC ;子程序,用于换行

MOV DL,10

MOV AH,02H

INT 21H

RET

CHANGE\_LINE ENDP

CODES ENDS

END START

**第五套**

DATAS SEGMENT

;此处输入数据段代码

INPUT\_REMIND DB 'Please input a number string ',10,'$'

GOOD\_NUM DB 10,'the good score is ','$'

BAD\_NUM DB 10,'the bad score iS ','$'

STUDENT DB 10,'the score of student is ','$'

BUF DB 100

DB ?

DB 100 DUP(?)

LONG EQU [BUF+1]

COUNT DB ?

X DW ? ;零时变量

GOOD DB 0 ;及格人数

BAD DB 0 ;未及格人数

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 DX,OFFSET INPUT\_REMIND

MOV AH,09H

INT 21H

MOV DX,OFFSET BUF

MOV AH,0AH

INT 21H

MOV DL,10

MOV AH,02H

INT 21H

MOV SI,2

MOV COUNT,0

TRUE:

MOV AL,100

MOV AH,[BUF+SI]

SUB AH,30H

MUL AH

MOV X,AX

INC SI

INC COUNT

MOV AL,10

MOV AH,[BUF+SI]

SUB AH,30H

MUL AH

ADD X,AX

INC SI

INC COUNT

MOV AL,1

MOV AH,[BUF+SI]

SUB AH,30H

MUL AH

ADD X,AX

INC SI

INC COUNT

INC SI

INC COUNT

MOV BX,X

CMP BX,60

JAE GOOD\_SCORE

JB BAD\_SCORE

GOOD\_SCORE:

INC GOOD

JMP JUDGE

BAD\_SCORE:

INC BAD

JMP JUDGE

JUDGE:

MOV AL,COUNT

CMP AL,LONG

JB TRUE

MOV SI,2

MOV COUNT,0

JMP OUTPUT\_SCORE

OUTPUT\_SCORE:

MOV DX,OFFSET STUDENT

MOV AH,09H

INT 21H

MOV DL,[BUF+SI]

MOV AH,02H

INT 21H

INC SI

INC COUNT

MOV DL,[BUF+SI]

MOV AH,02H

INT 21H

INC SI

INC COUNT

MOV DL,[BUF+SI]

MOV AH,02H

INT 21H

INC SI

INC COUNT

INC SI

INC COUNT

MOV AL,COUNT

CMP AL,LONG

JB OUTPUT\_SCORE

JMP OUTPUT\_NUMBER

OUTPUT\_NUMBER:

MOV DX,OFFSET GOOD\_NUM

MOV AH,09H

INT 21H

MOV DL,GOOD

ADD DL,30H

MOV AH,02H

INT 21H

MOV DX,OFFSET BAD\_NUM

MOV AH,09H

INT 21H

MOV DL,BAD

ADD DL,30H

MOV AH,02H

INT 21H

JMP EXIT

EXIT:

MOV AH,4CH

INT 21H

CODES ENDS

END START