

一、单项选择题(本大题共 10 小题, 每小题 3 分, 共 30 分)在每小题列出的四个备选项中只有一个是符合题目要求的, 请将其代码填写在题后的括号内。错选、多选或未选均无分。

1.CPU 要访问的某一存储单元的实际地址称(C)

A. 段地址 B. 偏移地址 C. 物理地址 D. 逻辑地址

2.某存储单元的物理地址是 12345H, 可以作为它的段地址有(D)

A. 2345H B. 12345H C. 12340H D. 1234H

3.执行后使 BX=0 的同时也使 CF=0, OF=0 的指令是(A)

A. XOR BX, BX B. OR BX, BX C. AND BX, BX D. CMP BX, BX

4.循环控制指令 LoopNZ/LoopNE 控制循环继续执行的条件是(B)

A. CX \neq 0 且 ZF=1 B. CX \neq 0 且 ZF=0 C. CX \neq 0 或 ZF=1 D. CX \neq 0 或 ZF=0

5.在执行 DAA 指令, 当高四位 BCD 码校正时产生进位, 如要把此进位值送入 AH 中, 对这进位值的操作应是(C)

A. DAA 校正指令的功能已自动加在 AH 中

B. 进位值在 AF 中, 校正后根据 AF 内容再加在 AH 中

C. 进位值在 CF 中, 校正后根据 CF 内容再加在 AH 中

D. 进位值在 AL 最高位上, 校正后根据 AL 最高位内容再加在 AH 中

6. AND, OR, XOR, NOT 为四条逻辑运算指令, 下面的解释正确的是(C)

A. 指令 XOR AX, AX 执行后, AX 内容不变, 但设置了标志位

B. 指令 OR DX, 1000H 执行后, 将 DX 最高位置 1, 其余各位置 0

C. 指令 AND AX, 0FH 执行后, 分离出 AL 低四位

D. NOT AX, 执行后, 将 AX 清 0

7. 在执行下列指令时，需要使用段寄存器 DS 的指令是(D)

A. STOSW B. ADD AL, CL C. NEG BX D. INC DA[BX]

8. 无论 BH 中原有的数是奇数或偶数，若要使 BH 中的数一定为奇数，应执行的指令是(B)

A. ADD BH, 01H B. OR BH, 01H C. XOR BH, 01H D. TEST BH, 01H

9. 比较两个带符号的数 A、B，当 A=B 时程序转移，测试的条件为(A)

A. ZF=1 B. ZF=0 C. SF=1 D. SF=0

10. 下面各传送指令中，正确的是(C)

A. MOV [DI], [SI] B. MOV[DX+DI], AL

C. MOV WORD PTR [BX], 0100H D. MOV AL, BX

11. 汇编语言语句格式中对名字项的规定如下，请找出其中错误的说法(B)

A. 名字的第一个字符可以是大写英文字母及小写英文字母

B. 名字的第一个字符可以是字母、数字及、@、_

C. 名字的有效长度 ≤ 31 个字符

D. 在名字中不允许出现\$

12. 要实现使 BETA 的值为 56，应采用语句为(C)

A. BETA DB 56 B. BETA DB 56H C. BETA EQU 56 D. BETA EQU 56H

13. REPZ CMPSW 指令，重复执行的终止条件是(A)

A. CX=0 或 ZF=0

B. CX=0 且 ZF=0

C. CX=0 或 ZF=1

D. CX=0 且 ZF=1

14. 下面程序段执行后, AL 中的内容是(**B**)

MOV AX, 8833H

ADD AL, AH

DAA

A. 21 B. 21H C. 0C1H D. 0DDH

15. 比较两个带符号的数 A、B, 当 A=B 时程序转移, 测试的条件为(**A**)

A. ZF=1 B. ZF=0 C. SF=1 D. SF=0

16. 检查二个无符号数的关系, 若要实现 $AL \geq BL$ 时分支去 LOP1 处, 那么在“CMP AL, BL”指令后应跟的分支指令是(**C**)

A. JNC LOP1 B. JA LOP1 C. JC LOP1 D. JGE LOP1

17. 在下列串操作指令中, 同时使用源串和目的串地址指针的指令是(**D**)

A. STOSW B. LODSW C. SCASW D. CMPSW

18. 设 AL=0AH, 下列指令执行后能使 AL=05H 的是(**C**)

A. NOT AL B. AND AL, 0FH C. XOR AL, 0FH D. OR AL, 0FH

19. DA1 DW 'AB', 'CD', 'EF', 'GH'(**D**)

⋮

MOV AX, DA1+3

指令执行后 AX 中的内容是

A. 'EF' B. 'CD' C. 'BC' D. 'FC'

20. 使用 DOS 系统功能调用时, 使用的软中断指令是(**B**)

A. INT 21 B. INT 10H C. INT 16H D. INT 21H

二、简答题（每题 10 分，共 30 分）

1.简述循环程序常用的控制方法。

解： 技术控制：适用于循环次数已知，包括正计数与倒计时两种。采用循环指令 LOOP 和 JCXZ 实现，将最大循环次数置入 CX,最后用 LOOP 对 CX 减 1 并判断是否为 0。 条件控制：用于循环次数未知。需要用转移指令判断循环条件，转移指令可以指定目的标号来改变程序运行顺序目的标号指向一个重复执行的语句开始或结束，就构成了循环控制结构，循环体为标号语句至转移指令。 混合控制：包括开关控制和逻辑尺控制，前者适用于循环内部有一定规律分支，循环前置开关走向，结束前设置下一个开关方向。逻辑尺控制适用于内部分支不规律，用位来控制每次循环的操作。

计数控制：循环次数已知，每循环一次+、-1；常用 LOOP 指令实现

条件控制：循环次数未知，须根据条件控制循环；用条件转移指令实现

状态控制：根据事先设置或实时检测的状态来控制循环

2.比较参数传递的三种方法,并分析递归调用时为什么不能采用约定存储单元法来传递数据？

寄存器： 将入口参数和出口参数放在与约定的寄存器，数据传递速度快，编程方便，节

约内存单元，但是寄存器数量有限，适用较少的子程序 **约定存储单元：** 不占用寄存器、参数个数任意，数据都有独立的存储单元，但定义变量有难度。 堆栈：不占用寄存器和存储单元，参数放在公共堆栈区，处理后恢复，注意参数的位置断点保存和恢复。

参数传递的方法主要有约定寄存器法、约定存储法与堆栈法三种。

1.约定寄存器法

约定寄存器法直接利用 CPU 内部寄存器传送参数。在转向子程序之前，主程序将入口参数送入指定寄存器中；调用子程序后，子程序从指定寄存器引用这些参数。经加工处理的结果也放入指定的寄存器中，这样从子程序返回主程序后可以从指定的寄存器中获得处理结果。这种方法最简单，但由于寄存器资源有限，故仅适用传递参数较少的情况。

2.约定存储法

约定存储法是在存储器中专门指定一些单元存放入口参数和出口参数。在转子程序之前，主程序用数据传送指令将入口参数存入口参数的存储单元中。调用子程序后，子程序按照约定从入口参数存储单元中取出这些参数进行处理。子程序完成处理，将处理结果送入指定的出口参数存储单元中，返回后主程序可从这些特定存储单元中获取需要的处理结果。约定存储法传送速度比约定寄存器法要慢些，适合于传递参数较多的情况。

3.堆栈法

堆栈法使用堆栈来完成数据传递。在调用子程序之前将参数压入堆栈，转入子程序后通过 SP 指针获得压入堆栈的参数地址，从而将其取出。子程序的处理结果亦送入堆栈保存。返回主程序后再通过出栈指令取出结果。使用该方法时应明了堆栈的变化情况，特别注意参

数入栈与出栈顺序。

堆栈法适合于参数较多且子程序有嵌套、递归调用的场合。

3.简述外部中断和软件中断的中断过程。

中断是用以提高计算机工作效率、增强计算机功能的一项重要技术。**外部中断**：当中断源向 CPU 发出中断请求，CPU 接受请求在一定条件下暂停执行原程序进行中断处理，执行完中断服务程序后再返回断点继续执行原程序。**软件中断**：是一条 CPU 指令，用以自陷一个中断。由于软中断指令通常要运行一个切换 CPU 至内核态的子例程，它常被用作实现系统调用。

4.简述循环程序结构的一般组成。

初始化部分：在循环程序的开头，先执行。设置地址指针的初值、计数器的初值及程序中用到的某些寄存器和某些内存单元的初值，只执行一遍。 **工作部分**：从初始化部分设置的初值开始，动态地反复执行相同的操作，实现循环程序的功能。**修改部分**：对参加运算的数据或地址指针和记过单元地址指针修改 **控制部分**：按预定循环次数的执行和正常退出循环。

5.简述多分支程序实现的一般步骤与常用处理方法。

使用条件转移 JCC 指令和无条件专题 JMP 指令实现分支程序结构。获得条件变量-->对应转换为地址表入口偏移--> $BX=BX+入口偏移$ JMP[BX]跳转到相应位置。常用地址表的方法。无法使用地址表，只能通过多条条件转移指令实现。

6.简述完整子程序结构所包含的内容。

子程序也是一段程序，由文字说明和子程序本身构成。一般由子程序名、子程序的功能。子程序选用的寄存器名和存储单元分配、子程序入口参数、子程序出口参数。

三.编写程序，计算两个 64 位数的和，不考虑和的进位。（20 分）

由于 64 位数占用的寄存器比较多，故适合于用约定存储单元法来进行处理。

程序中 64 位加数 M1 和 M2 存放在两个双字单元内。由于不考虑和的进位，所以和也可用两个双字单元来存放。

程序如下：

DATA SEGMENT

NUM1 DD 1234H, 12345678H: M1 为 123456781234H

MM2 DD 4321H, 87654321H: M2 为 876543214321H

SUM DD?, ?

DATA ENDS

STACK SEGMENT PARA STACK 'STACK'

DB 100 DUP(?)

STACK ENDS

CODE SEGMENT

ASSUME CS: CODE, SS: STACK, DS: DATA, ES: DATA

MAIN PROC: 主程序

BEGIN: PUSH DS

MOV AX, 0

PUSH AX

MOV AX, 0

MOV DS, AX

MOV ES, AX

CALL ADD64: 调用子程序 ADD64

RET

MAIN ENDP

: 子程序名称: ADD64

: 子程序功能: 两个 64 位数的相加

: 输入参数: M1 一被加数的地址

: M2 一加数地址

: 输出参数: SUM 一和数地址

ADD64 PROC

PUSHF

PUSH AX: 保护现场

PUSH BX

PUSH CX

LEA BX, M1: Bx 指向被加数单元

MOV CX, 4

```

CLC
AGAIN:  MOV Ax, [Bx]; 取被加数的相应字— Ax
        ADc Ax, [Bx+8]; 与加数相应字相加
        MOV  [Bx+10H], Ax; 保存和
        INC BX
        INC BX
        LOOP AGAIN
        POP Cx: 恢复现场
        POP BX
        POP AX
        POPF
        RET
ADD64 ENDP
CODE ENDS
END BEGIN

```

四.编写程序，将一个十六位二进制数转换为 ASCLL 码，并以十进制数形式在屏幕上输出。（20 分）

```

DATA    SEGMENT
BINARY  DW  0110110000001100B
BUF      DW   9 DUP
DATA     ENDS
STACK1  SEGMENT  PARA STACK 'STACK'
        DB 100H   DUP
STACK   ENDS
CODE    SEGMENT
        ASSUME CS: CODE,  DS:  DATA
START   PROC  FAR;
        PUSH  DS
        MOV   AX,  0
        PUSH  AX
        MOV   AX,  DATA
        MOV   DS,  AX
        MOV   CX,  BINARY ;CX
        CAALL CONV ;
        CALL  DISPLAY ;
        RET
START   ENDP

```

;子程序名称: CONV

;子程序功能: 将 CX 中十六位二进制数转换为十进制数

```

; 入口参数:      CX
;调用子程序:      CHANGE
CONV  PROC  NEAR
        PUSH  AX
        PUSH  BX
        PUSH  DX
        PUSH  SI
        LEA   BX, BUF
        MOV  AL,  0AH
        MOV  [BX], AL;
        INC  BX
        MOV  AL,  CH;
        OR   AL,  AL
        JNS  PLUS;
        NEG  CX;
        MOV  AL, '-'
        MOV  [BX], AL ;
        JMP  GOON
PLUS:   MOV  AL, '+'
        MOV  [BX], AL;
GOON:   INC  BX
        MOV  SI,  10000;
        CALL CHANGE
        MOV  SI,  1000;
        CALL CHANGE
        MOV  SI,  100;
        CALL CHANGE
        MOV  SI,  10;
        CALL CHANGE
        MOV  AL, CL;
        ADD  AL,  30H
        MOV  [BX], AL
        INC  BX
        MOV  AL, '$';
        MOV  [BX], AL
        POP  SI
        POP  DX
        POP  BX
        POP  AX

```



```

        RET
CONV     ENDP
;子程序名称:  CHANGE
;子程序功能:  统计 CX 中所包含权 (在 SI 中) 的个数
;              并把个数转换为 ASCH 码 , 存入缓冲区中
; 入口参数:   CX
; 调用子程序:  CHANGE
CONV  PROC  NEAR
        PUSH  AX;
        PUSH  BX
        PUSH  DX
        PUSH  SI
        LEA   BX, BUF
        MOV  AL,  0DH
        MOV  [BX], AL;
        INC  BX
        MOV  AL, CH;
        OR   AL, AL
        JNS  PLUS;
        NEG  CX;
        MOV  AL, '-'
        MOV  [BX], AL;
        JMP  GOON
PLUS:    MOV  AL, '+'
        MOV  [BX], AL;
CHANGE  ENDP
;子程序名称:  DISPLAY
;子程序功能:   输出以$结尾的一个字符串
;调用子程序:  系统 09 号功能
DISPLAY PROC NEAR
        PUSH AX
        MOV DX,OFFSET BUF
        MOV AH,9
        INT 21H
        POP AX
        RET
DISPLAY ENDP
        CODE ENDS
        END START

```

五.编写程序，将输入的大写字母转换为小写字母。（20 分）

```
DATAS SEGMENT
a db "input",0dh,0ah,"$"
DATAS ENDS
CODES SEGMENT
ASSUME CS:CODES,DS:DATAS
START:
mov ax,DATAS
mov ds,ax
mov dx,offset a ;定义数据段
mov ah,9h
int 21h ;输出“input”，并且让下一位所要显示的显示在下一行
mov ah,1
int 21h ;输入
mov dl,al ;转存输入
sub dl,20h
mov ah,2
int 21h ;输出此时的 DL 中的数值（为 ASCII 码）
mov ax,4c00h
int 21h
CODES ENDS
END START
```

六.对两个整数变量 A 和 B，编写程序完成下列功能：

- 1、若两个数中有一个是奇数，则将奇数送入 A 单元中，偶数送入 B 单元中。
- 2、若两个数均为奇数，则两个数分别加 1，并存入原变量中。
- 3、若两个数均为偶数，则两变量不变。（20 分）

```
dseg segment
    a    dw  ?
    b    dw  ?
dseg ends
```

```

cseg    segment

mainproc  far

assume   cs:cseg,ds:dseg

start:push ds

        sub     ax,ax

        push ax

        mov     ax,dseg
        mov     ds,ax

begin:   mov     ax,a
        mov     bx,b
        xor     ax,bx
        test    ax,0001
        jz      class
        test    bx,0001
        jz      exit
        xchgbx,a
        mov     b,bx
        jmp     exit

class:   test    bx,0001
        jz      exit
        inc     b
        inc     a

exit: ret

mainendp

cseg     ends

        end     start

```