**2017-2018学年第2学期《汇编语言程序设计》考试试卷**

**B卷 闭卷 考试时间： 2018年 月 日**

**专业 班级 学号 学生姓名**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题号 | 一 | 二 | 三 | 四 | 五 | 六 | 七 | 总分 | 核对人 |
| 题分 | 10 | 10 | 10 | 20 | 10 | 20 | 20 | 100 |  |
| 得分 |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| 得分 | 评卷人 |
|  |  |

**一、选择题（共10分，每空1分）**

1. 指令MOV AX，OFFSET [BX+3] 错误的原因是\_\_\_\_\_C\_\_\_\_\_\_\_。

A. 源操作数不应为变址寻址 B. 两操作数数据类型不明确

C. OFFSET后不能跟加方括号的寄存器符号 D. OFFSET后不能跟变量和标号

2. 指令MOV AX，BL错误的原因是\_\_\_\_\_\_C\_\_\_\_\_\_\_\_。

A. 源操作数表示的数据范围太小 B. 目的操作数表示的数据范围太小

C. 两操作数类型不一致 D. 不允许往AX传送8位字节数据

3. 下列指令正确的语句是\_\_\_\_\_\_\_\_\_\_D\_\_\_\_\_\_\_\_\_\_\_\_。

A. ADD AX，BX+3 B. MOV CS, AX

C. MOV DS, 1000H D. LEA BX, [SI+3]

4. 指令PUSH AL错误的原因是\_\_\_\_\_\_\_C\_\_\_\_\_\_\_\_\_。

A. 源操作数不能是寄存器 B. 源操作数不能是字类型

C. 源操作数不能是字节类型 D. 以上说法都不正确

5. 指令 ADD CX，[EAX\*20+13]错误的原因是\_\_\_\_\_\_C\_\_\_\_\_\_\_。

A. 不能用EAX作寄存器间接寻址 B. 不能用EAX作为变址寻址

C. 变址寻址比例因子不能是20 D. 变址寻址位移量不能为奇数

6. 可以用来访问外部设备寄存器的指令语句是\_\_\_\_\_\_\_C\_\_\_\_\_\_\_\_\_。

A. MOV 45H， AL B.　ADD BX, [BX]

C. IN AL, 60H D. ROL AX， 4

7. 设（BX）=7018H， (CX)=2018H， 则执行 ADD BX，CX后，有\_\_\_D\_\_\_\_。

A. SF=0，OF=0 B. SF=0，OF=1

C. SF=1，OF=0 D. SF=1，OF=1

8.设 (BX)=2018H，CF=0 则执行循环右移指令 ROR BX, 4后，有\_\_\_\_B\_\_\_\_\_\_。

A. （BX）= 8201H， CF=0 B. (BX) = 8201H，CF=1

C. （BX）= 0201H， CF=1 D. (BX) = 0201H，CF=0

9. 设BUF为字节类型变量，则下列指令语句中没有语法错误的是\_\_\_\_\_\_B\_\_\_\_\_\_。

A. MOV BUF, BX B. ADD BUF，[BX]

C. MOV BUF[BX]，AX D. SUB BUF[BX]，1200H

10. 指令IDIV CX执行后，AX中存放的是\_\_\_\_A\_\_\_\_\_。

A. 商 B. 余数 C.被除数 D.除数

|  |  |
| --- | --- |
| 得分 | 评卷人 |
|  |  |

**二、填空题（共10分，每空1分）**

1. 设 (BX)=5678H，(EDX)=44551111H，则执行下列指令语句后：

PUSH BX

PUSH EDX

POP BX

POP EDX

(EDX)=\_\_\_\_56784455H\_\_\_\_\_， （BX）=\_\_\_\_1111H\_\_\_\_\_。

2. 在实方式下，(DS)=2000H， (SS)=5000H，(BX)=100H，(BP)=300H， 则指令MOV AX, [BX]中源操作数的物理地址是\_\_\_20100H\_\_\_。

3. 设数据段定义如下：

DATA SEGMENT USE16

A1 DW -1，1，9

B1 DB 1

C1 DB 2

DATA　ENDS

执行下列语句之后:（AX）=\_\_\_01FFH\_\_\_，(BX)=\_\_\_0201H\_\_\_\_。

MOV BX，WORD PTR B1

MOV AX，A1+1

4. CPU执行指令：CALL DISPLAY的主要操作是(DISPLAY是一个NEAR类型子程序)：

\_\_将（IP）压栈，然后将DISPLAY的入口地址赋给IP\_\_。

5. 在模块化程序设计中，若模块1需要调用模块2中定义的子程序PRINT (PRINT为FAR类型)，则在模块1中的声明是\_\_EXTRN PRINT:FAR\_\_，模块2中的声明是\_PUBLIC PRINT\_。

6. 从端口号是60H的设备中读取字节数据的指令是\_\_\_IN AL, 60H\_\_，往71H端口输出字节数据的指令是\_\_\_OUT 71H, AL\_\_\_。

|  |  |
| --- | --- |
| 得分 | 评卷人 |
|  |  |

**三、已知数据段定义如下，请回答下面的问题（共10分）**

DATA SEGMENT USE16

BUF DW A1

B1 DW 2018H

E1 DB 2 DUP(‘3’, 5)

A1 DB ‘PEN’， ‘$’

C1 EQU $-A1

DATA ENDS

|  |  |
| --- | --- |
| 低地址 BUF 00  1. 以字节为单位，画出该数据段的数据存储示意图。（4分）  2.变量B1、E1的偏移地址值是多少？C1的值是多少？(2分)  **02H, 04H, 04H**  3. 假设DS已正确赋值，执行下列指令后，各寄存器的内容。(2分)  LEA BX, A1 （BX）=\_\_\_\_\_\_**0008H**\_\_\_\_\_\_\_  MOV AL, E1 (AL)=\_\_\_\_\_\_\_**33H**\_\_\_\_\_\_\_\_\_  MOV [BX]，AL  LEA DX， A1  MOV AH, 9  INT 21H  4执行完上述语句序列后，屏幕显示什么？(2分)  **显示 3EN**  01  B1 02  03  E1 04  05  06  07  A1 08  09  0A  0B  0C  0D  高地址 | 08H |
| 00H |
| 18H |
| 20H |
| 33H |
| 05H |
| 33H |
| 05H |
| ‘P’ |
| ‘E’ |
| ‘N’ |
| ‘$’ |
|  |
|  |
|  |

|  |  |
| --- | --- |
| 得分 | 评卷人 |
|  |  |

**四、简答题（共20分，每小题5分）**

1. 基于窗口的Windows程序中窗口消息处理程序主要由哪四个部分组成？

**主程序、窗口主程序、消息处理程序、用户处理程序。**

2.写出计算（X\*10-Y）/ 8并将结果保存在字变量Z中的程序段（不考虑溢出），已知字变量X，Y和Z定义如下：

X DW 60H

Y DW -50

Z DW ?

**MOV AX, 10**

**IMUL X**

**SUB AX, Y**

**SBB DX, 0**

**MOV BX, 8**

**IDIV BX**

**MOV Z, AX**

3. 已知 BUF是字节类型的变量，BUF的定义： BUF DB 2，3，4，-5，请按照要求，写出完成相应功能的程序段。

（1）用寄存器间接寻址方式，将 BUF+2的字内容送入BX

**LEA SI, BUF+2**

**MOV BX, [SI]**

（2）用变址寻址方式，将 BUF+2中的字内容送入BX

**MOV SI, 2**

**MOV BX, WORD PTR BUF [SI]**

4.编写宏指令CHANGE A, B，其中A, B为字类型变量，该宏指令交换A，B的值且不影响任何寄存器的值。

**CHANGE MACRO A, B**

**PUSH AX**

**XCHG AX, A**

**XCHG AX, B**

**XCHG AX, A**

**POP AX**

**ENDM**

|  |  |
| --- | --- |
| 得分 | 评卷人 |
|  |  |

**五、指令填空（共10分，每空1分）**

1.在数据段中存放有若干学生的学号、成绩信息，一个学生的存储定义是：

STUDENT DW 1111; 学生学号

DB 100; 成绩

下列程序完成统计计算学生的平均成绩，并将平均成绩保存在字类型变量AVG中，完善下列程序。

.386

DATA SEGMENT USE16

STUDENT DW 1111

DB 98

STUDENT1 DW 1112

DB 100

…………

STUDENTN DW 2222

DB 78

N=\_\_**($ - STUDENT)**\_\_\_/ 3 (或 **$** / 3)

AVG DW 0

DATA ENDS

CODE SEGMENT USE16

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

START: MOV AX, DATA

MOV DS, AX

**LEA SI, \_ STUDENT \_**

MOV CX, N

MOV AX, 0

NEXT: MOVZX DX, BYTE PTR [SI+2]

**\_ADD AX, DX\_\_**

ADD SI, 3

LOOP NEXT

**MOV BX, N\_**

MOV DX, 0

DIV BX

**MOV AVG, AX\_\_**

MOV AH, 4CH

INT 21H

CODE ENDS

END START

2. 下面是一个接管键盘中断(16H的功能号0和10H)的中断服务程序并驻留内存，在程序返回DOS操作系统后，输入键盘上的小写字母时都变成了大写字母。

.386

CODE SEGMENT USE16

ASSUME CS:CODE, SS:STACK

OLD\_INT DW ?, ?

NEW\_INT PROC

CMP AH, 0

JE L1

CMP AH, 10H

JE L1

JMP DWORD PTR OLD\_INT

L1: PUSHF

**\_CALL \_ DWORD PTR OLD\_INT \_**

CMP AL, 'a'

**\_JB L2\_\_\_\_\_**

CMP AL, 'z'

JA L2

SUB AL, 20H

L2: **\_IRET\_\_\_\_\_**

NEW\_INT ENDP

START: XOR AX, AX

MOV DS, AX

MOV AX, DS:[16H\*4]

MOV OLD\_INT, AX

MOV AX, DS:[16H\*4+2]

MOV OLD\_INT+2, AX

CLI

**MOV DS:[16H\*4], OFFSET NEW\_INT**

**MOV DS:[16H\*4+2], CS \_\_**

STI

MOV DX, OFFSET START+15

MOV CL, 4

SHR DX, CL

ADD DX, 10H

MOV AL, 0

MOV AH, 31H

INT 21H

CODE ENDS

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

END START

|  |  |
| --- | --- |
| 得分 | 评卷人 |
|  |  |

**六、程序分析（共20分）（每个问题5分）**

阅读下列程序，回答问题。

.386

DATA SEGMENT USE16

STU DB 'ZHANGSAN', 0, 0

DB 'C'

STU1 DB 'LISI', 0, 5 DUP(0)

DB 'B'

STU2 DB 'WANGWU', 0, 3 DUP(0)

DB 'A'

STU3 DB 'ZHAOLIU', 0, 0, 0

DB 'D'

N = ($ - STU) / 11

HINT DB 'PLEASE INPUT A STUDENT NAME: $'

BUF DB 50

DB ?

DB 50 DUP(0)

CRLF DB 0DH, 0AH, '$'

NOTFIND DB 'NOT FOUND$'

DATA ENDS

STACK SEGMENT USE16 STACK

DB 200 DUP(0)

STACK ENDS

CODE SEGMENT USE16

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

START: MOV AX, DATA

MOV DS, AX

LEA DX, HINT

MOV AH, 9

INT 21H

LEA DX, BUF

MOV AH, 10

INT 21H

LEA DX, CRLF

MOV AH, 9

INT 21H

MOV BL, BUF+1

MOV BH, 0

CMP BX, 0

JE EXIT

MOV BUF+2[BX], 0; -------------①

LEA BX, STU

MOV CX, N

LOP1: PUSH OFFSET BUF+2

PUSH BX

CALL STRCMP1

CMP AX, 1

JE FIND

ADD BX, 11

LOOP LOP1

LEA DX, NOTFIND

MOV AH, 9

INT 21H

EXIT: MOV AH, 4CH

INT 21H

FIND: MOV DL, [BX+10]

MOV AH, 2

INT 21H

JMP EXIT

STRCMP1 PROC

PUSH BP

MOV BP, SP;----------------②

PUSH SI

PUSH DI

PUSH BX;---------------------③

MOV SI, [BP+6]

MOV DI, [BP+4]

MOV AX, 0

LL1: MOV BL, [SI]

CMP BL, [DI]

JNE LL2

INC SI

INC DI

CMP BL, 0

JE FIND1

JMP LL1

FIND1: MOV AX, 1

LL2: POP BX

POP DI

POP SI

POP BP

RET 4

STRCMP1 ENDP

CODE ENDS

END START

1. 该程序完成什么功能？

**从键盘输入一个学生的姓名，然后在学生成绩表中查找该学生，若找到则显示该学生成绩的等级（A、B、C、D等），否则显示“NOT FOUND”。**

2. 子程序STRCMP1完成什么功能？入口参数和出口参数分别是什么？

**比较2个字符串是否相同，字符串以0结尾。2个入口参数保存在堆栈中，表示2个字符串的首地址（偏移地址）；出口参数为（AX），1表示相同，0表示不相同。**

3. 若漏写语句①，该程序在输入’LISI’后，会输出什么结果？

**一般情况下会输出“NOT FOUND”，因为子程序会一直比较2个字符串，直到在BUF+2开始的内存中遇到0。**

4. 画出程序运行完语句②时的堆栈示意图，并标明此时BP指向的存储单元的位置；

**（BP）的值 （2字节） 堆栈低地址，（BP）=（SP）指向该单元**

**STRCMP1的偏移地址（2字节）**

**（BX）的值 （2字节）**

**BUF偏移地址 + 2 （字节） 高地址**

5. 若漏写语句③，在输入’LISI’后，程序的运行结果是什么？原因是什么？

**程序会崩溃，因为STRCMP1中POP语句比PUSH语句多，导致堆栈被破坏，从而STRCMP1不能返回到主程序的正确位置。**

|  |  |
| --- | --- |
| 得分 | 评卷人 |
|  |  |

**七、程序设计（20分）**

编写一个完整的实方式下程序，实现如下功能：在从键盘输入一个长度小于10的字符串到BUF字节存储区（假设输入的字符串中没有重复的字符），对输入的字符串，根据字符的ASCII码的大小，计算每个字符在整个字符串中的次序，然后在屏幕上显示输出每个字符及其次序。例如输入字符串：'string'，则输出格式为：

s: 5

t: 6

r: 4

i: 2

n: 3

g: 1

**要求：**(1) 画出程序流程图；

(2) 程序完整（包括堆栈段、数据段、代码段定义等），至少给出2条必要的注释；

(3) 用子程序RANK实现计算一个字符的次序。

RANK：开始

主程序：开始

返回: (AL)为字符的次序

字符串首址→SI

0→BX, 表示字符的编号

Y

N

取需考虑的字符到AH:

[BX+SI] → AH；保护SI, 1→AL

说明: (BX)为字符编号

(SI)指向以0DH结尾的字符串

（SI）+1 → SI

（AL）+1 → AL

N

Y

Y

[SI] < (AH) ?

恢复SI

[SI] = 0DH ?

N

结束

（BX）+1 → BX

输出该字符的次序

RANK

得到字符的次序

(BX)>=

字符串长度？

输入字符串→BUF

输出回车换行

.386

;----------------------------

STACK SEGMENT STACK USE16

DB 200 DUP(0)

STACK ENDS

;----------------------------

DATA SEGMENT USE16

BUF DB 10, ?, 10 DUP(0)

RESULT DB '?: ?', 0AH, 0DH, '$'

MSG DB 'Input a string: $'

DATA ENDS

;------------------------------

CODE SEGMENT USE 16

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

BEGIN: MOV AX, DATA

MOV DS, AX

LEA DX, MSG

MOV AH, 9

INT 21H

LEA DX, BUF

MOV AH, 10

INT 21H

LEA DX, RESULT+4

MOV AH, 9

INT 21H ;输出回车换行

;;

LEA SI, BUF+2 ;(SI)为字符串首地址,以0DH结尾

MOV BX, 0 ;(BX)表示第几个字符（>=0）

L1: CMP BL, BUF+1

JAE L2

CALL RANK

ADD AL, '0' ;返回 (AL)为第(BX)个字符在整个字符串中的次序

MOV RESULT+3, AL

MOV AL, [BX+SI]

MOV RESULT+0, AL

LEA DX, RESULT

MOV AH, 9

INT 21H

INC BX

JMP L1

;;

L2: MOV AH, 4CH ;exit

INT 21H

;-----------------------------

;功能：计算第(BX)(>=0) 个字符在字符串(首地址为(SI),以0DH结尾)中的次序

;返回：(AL)为第(BX)个字符在整个字符串中的次序 (>=1)

RANK PROC

MOV AL, 1

MOV AH, [BX+SI]

PUSH SI

R1: CMP BYTE PTR [SI], 0DH

JZ R3

CMP AH, [SI]

JBE R2

INC AL ;ASCII码比(AH)小的字符个数

R2: INC SI

JMP R1

R3: POP SI

RET

RANK ENDP

CODE ENDS

END BEGIN