CSE 2312 Computer Organization & Assembly Language Programming

QUIZ5

Student Name: ____.

Student ID: ____.

TRUE OR FALSE (2pts per)

Q:	1	2	3	4	5	6	7	8	9	10
T/F	T	T	T	F	F	F	F	F	F	T

1. The translation from source procedure to object module represents a change of level while the linking process does not represent a change of level. T

2. A linker reads five modules, whose lengths are 200, 800, 600, 500, 700 words. If they are loaded in that order, the relocation constants are 0, 200, 1000, 1600, 2100. T

3. We create a macro X. Inside it, call X. In that case it will fall into an infinite loop. T

4. The value of AX is 707 after executing **MOV AX, 258** and then **ADDB AH, AL** (should be 770) F

5. If the program counter has value 0, then it refers to the absolute memory address zero. F

6. The assembler is a translator while the compiler is not. F

7. Macros can call other macros, but can not call themselves. F

8. Macro expansion occurs during the execution of the program. F

9. An expert on machine language programming may tell whether or not any macros were involved from the corresponding machine code. F

10. Assembly Counter used during the assembly time to point where to put the next instruction in the memory. On the other hand, Program Counter keeps track of which instruction to execute next during the run time. T

Multiple Choices (5 pts each)

(The following questions may have one and more correct answer. Pick all correct answers.)

1. **Which of the following is true? [a, b, c]**

(a) Translator is the programs that convert a user's program written in some language to another language

(b) Translation is used when a processor (either hardware or an interpreter) is available for the target language but not for the source language.

(c) Correct translation will give precisely the same results as the execution of the source program

(d) The Assembly Language layer is implemented by interpretation rather than by translation

2. **EQU is pseudo-instruction that equates the label to the expression in the operand field. How many passes are required for the following program be assembled? [d]**

```
P EQU Q
Q EQU R
R EQU S
S EQU 4
```

(a) 1
(b) 2
(c) 3
(d) 4

3. **Which of the following is true for the assembly language? [a, b, c, d]**

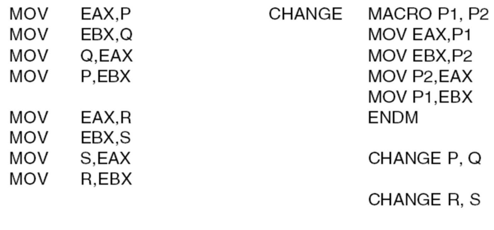
(a) Writing a program in assembly language often takes much longer than writing the same program in a high-level language.

(b) It often takes much longer to debug and is much harder to maintain.

(c) An expert assembly language programmer can often produce code that is much smaller and much faster than a high-level language programmer can.

(d) It can help us to fully access to the hardware, something usually impossible in high-level languages.

4. Which of the following is true? [b, c]



```
MOV   EAX,P              CHANGE   MACRO P1, P2
MOV   EBX,Q                       MOV EAX,P1
MOV   Q,EAX                       MOV EBX,P2
MOV   P,EBX                       MOV P2,EAX
                                  MOV P1,EBX
MOV   EAX,R                       ENDM
MOV   EBX,S
MOV   S,EAX                       CHANGE P, Q
MOV   R,EBX
                                  CHANGE R, S
```

(a) (b)

(a) P1 and P2 are actual parameters of the macro CHANGE
(b) P1 and P2 are formal parameters of the macro CHANGE
(c) R and S are actual parameters of the macro CHANGE
(d) P and Q are formal parameters of the macro CHANGE

5. Which of following is true? [a, b, c, d]

(a) The stack pointer, SP, should always contain an even number.
(b) The PUSH instruction decreases the stack pointer by 2.
(c) The POP command retrieves the value, and increments SP by 2.
(d) The stack is filled up from high addresses to low addresses.

6. Which of the following will cause errors in Assembly Language? [a, b, d, e]

(a) A symbol has been defined more than once.
(b) An opcode is not supplied with enough operands.
(c) An octal number contains a 7.
(d) An opcode is supplied with too many operands.
(e) The END statement is missing.

Computation and Short Answer (10pts each)

(**Please give detailed computation process for your final results!**)

1. Please fill the following symbol table after the following Pentium 4 statements have been encountered. The first statement is assigned to address 8000.

Label	Instruction	Length	Instruction Location Counter
Everest:	POP BX	1 byte	**8000**
K2:	PUSH BP	1 byte	**8001**
WHITNEY:	MOV BP, SP	2 byte	**8002**
MCKINLEY:	PUSH X	3 byte	**8004**
FUJI:	PUSH SI	1 byte	**8007**
KIBO:	SUB SI, 300	3 byte	**8008**

2. For a certain program, 1% of the code accounts for 50% of the execution time. Compare the following three strategies with respect to programming time and execution time. Assume that it would take 100 man months to write it in C, and that assembly code is 10 times slower to write and five times more efficient.

1) Entire program in C
2) Entire program in assembler.
3) First all in C, then the key 1% in assembly.

Solution:

1) 100 man month, execution time = X (let)
2) 100 * 10 = 1000 man month, execution time = X/5
3) 100 + 1*10 = 110 man month, execution time = 0.5X + (0.5/5)X = 0.6X.

3. Compute the hash code for each of the following symbols by adding up the letters and taking the result module the hash table size.

els, jan, jelle, maaike

(Here, a=1, e = 5, j=10, l=12, m=13, n=14, s=19)

1) The hash table has 19 slots, numbered 0 to 18. Does each of them generate unique hash code? If not, how to deal with the collision?
2) The hash table has 20 slots, numbered 0 to 19. Does each of them generate unique hash code? If not, how to deal with the collision?

Solution:

1) els = (5 + 12 + 19) mod 19 = 17

jan = (10 + 1 + 14) mod 19 = 6

jelle = (10 + 5 + 12 + 12 + 5) mod 19 = 6

maaike = (13 + 1 + 1 + 9 + 11 + 5) mod 19 = 2

jan and jelle hash to the same value. We can maintain a linked list in the slots that contain all the elements. (e,g. 6 will contain both jan and jelle).

2) els = (5 + 12 + 19) mod 20 = 16

jan = (10 + 1 + 14) mod 20 = 5

jelle = (10 + 5 + 12 + 12 + 5) mod 20 = 4

maaike = (13 + 1 + 1 + 9 + 11 + 5) mod 20 = 0

We obtain the unique hash code.

4. Suppose AX=258 and BX=2, Please give the values of AX, AH, AL after executing each instruction:

```
ADD AH, AL    ! 1   AX=3*256+2
SUB AX, BX    ! 2   AX=3*256
SUB AH, BL    ! 3   AX=1*256
ADD AX, 1     ! 4   AX=257
```

5. Suppose that CS = 50, DS = 2000, and IP = 20.

a. What is the absolute address of the next instruction?

b. If MOV AX, (2) is executed, which memory word is loaded into AX?

Solution:

a. The code segment begins at address 800, so the next instruction is at 820.

b. The word referenced is at 2000 × 16 + 2, or 32,002.