|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **[40] Answers for question 1 [Put √ for the correct answer]** | | | | | | | | | | |
|  | **A** | **B** | **C** | **D** |  |  | **A** | **B** | **C** | **D** |
| 1 |  |  |  |  | 21 |  |  |  |  |
| 2 |  |  |  |  | 22 |  |  |  |  |
| 3 |  |  |  |  | 23 |  |  |  |  |
| 4 |  |  |  |  | 24 |  |  |  |  |
| 5 |  |  |  |  | 25 |  |  |  |  |
| 6 |  |  |  |  | 26 |  |  |  |  |
| 7 |  |  |  |  | 27 |  |  |  |  |
| 8 |  |  |  |  | 28 |  |  |  |  |
| 9 |  |  |  |  | 29 |  |  |  |  |
| 10 |  |  |  |  | 30 |  |  |  |  |
| 11 |  |  |  |  | 31 |  |  |  |  |
| 12 |  |  |  |  | 32 |  |  |  |  |
| 13 |  |  |  |  | 33 |  |  |  |  |
| 14 |  |  |  |  | 34 |  |  |  |  |
| 15 |  |  |  |  | 35 |  |  |  |  |
| 16 |  |  |  |  | 36 |  |  |  |  |
| 17 |  |  |  |  | 37 |  |  |  |  |
| 18 |  |  |  |  | 38 |  |  |  |  |
| 19 |  |  |  |  | 39 |  |  |  |  |
| 20 |  |  |  |  | 40 |  |  |  |  |
| **[10] Answers for question 2** | | | | | | | | | | |
|  | **V/I** |  | | |
| **1** |  |
| **2** |  |
| **3** |  |
| **4** |  |
| **5** |  |
| **6** |  |
| **7** |  |
| **8** |  |
| **9** |  |
| **10** |  |

|  |  |  |
| --- | --- | --- |
| **Cairo University** | **CMPA201** | **Total:120 Points** |
| **Faculty of Engineering** | **Microprocessor Systems** | **2016-2017** |
| **Computer Eng. Department** | **Final Exam [Answer Sheets]** | **Two Hour** |

**Please fill in your answers in the following eight pages.**

**[6] Answers for question 3**

|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
|  |
|  |
| 3 |  |
| 4 |  |

**[4] Answers for question 4**

**[60] Answers for question 5**

**[A-10]**

|  |  |
| --- | --- |
| ; Count the number of times that the given  ; character code occurs in the specified string  ;  ; Entry: AL - character code to count  ; DS:SI - pointer to zero terminated string  ; Exit: AX - count of times character occurs  ; Uses: AX modified, all else preserved |  |

**[B-10]**

|  |  |
| --- | --- |
|  |  |

**[C-10]**

|  |  |
| --- | --- |
|  |  |
| **[D-5]** | **[E-5]** |
|  |  |
| **[F-5]** | **[G-5]** |
|  |  |
| **[H-5]** | **[I-5]** |
|  |  |

**Extra Page**

|  |  |  |
| --- | --- | --- |
| **Cairo University** | **CMPA201** | **Total:120 Points** |
| **Faculty of Engineering** | **Microprocessor Systems** | **2016-2017** |
| **Computer Eng. Department** | **Final Exam** | **Two Hour** |

**This is an open-book, open notes exam. All electronic devices - Except calculators - are forbidden.  
Make any reasonable assumptions (if necessary)  
Answer the following questions at the answer sheets**

**Q1 - [40] Choose the correct answer**

|  |  |  |
| --- | --- | --- |
| 1. **The instruction, MOV AX, 0005H belongs to the address mode** | | |
| a) register | | b) direct |
| c) immediate | | d) register relative |
| 1. **If the offset of the operand is stored in one of the index registers, then it is** | | |
| a) based indexed addressing mode | | b) relative based indexed addressing mode |
| c) indexed addressing mode | | d) none of the mentioned |
| 1. **The instruction that enables subtraction with borrow is** | | |
| a) DEC | | b) SUB |
| c) SBB | | d) None of the mentioned |
| 1. **In general, the source operand of an instruction can be** | | |
| a) memory location | | b) register |
| c) immediate data | | d) all the above |
| 1. **The instruction, CMP to compare source and destination operands it performs** | | |
| a) addition | | b) subtraction |
| c) division | | d) multiplication |
| 1. **Both the operands source and destination of an instruction cannot be** | | |
| a) register, register | | b) immediate data, register |
| c) register, immediate data | | d) immediate data, memory location |
| 1. **The coded object modules of the assembled program are present in** | | |
| a) .ASM file | | b) .OBJ file |
| c) .EXE file | | d). OBJECT file |
| 1. **The advantages of assembly level programming compared to machine language is** | | |
| a) flexibility of programming is more | | b) chances of error are less |
| c) debugging is easy | | d) all of the mentioned |
| 1. **As the storing of data words onto the stack is increased, the stack pointer is** | | |
| a) incremented by 1 | | b) decremented by 1 |
| c) incremented by 2 | | d) decremented by 2 |
| 1. **The process of storing the data in the stack is called ……… the stack.** | | |
| a) pulling into | | b) pulling out |
| c) pushing into | | d) popping into |
| 1. **The instruction that performs logical AND operation and the result of the operation is not available is** | | |
| a) AAA | | b) AND |
| c) TEST | | d) XOR |
| 1. **The instruction that is used as prefix to an instruction to execute it repeatedly until the CX register becomes zero is** | | |
| a) LOOP | | b) REP |
| c) CMPS | | d) STOS |
| 1. **The instruction that multiplies the content of AL with a signed immediate operand is** | | |
| a) MUL | | b) SMUL |
| c) IMUL | | d) none of the mentioned |
| 1. **The stack pointer register contains** | | |
| a) address of the stack segment | | b) pointer address of the stack segment |
| c) offset of address of stack segment | | d) data present in the stack segment |
| 1. **The register or memory location that is pushed into the stack at the end must be** | | |
| a) popped off last | | b) pushed off first |
| c) popped off first | | d) pushed off last |
| 1. **The number of PUSH instructions and POP instructions in a subroutine must be** | | |
| a) PUSH instructions must be greater than POP instructions | | b) POP instructions must be greater than PUSH instructions |
| c) both must be equal | | d) instructions may be any kind |
| 1. **Procedures are also known as** | | |
| a) macros | | b) segment |
| c) subroutines | | d) none |
| 1. **For passing the parameters to procedures using the PUBLIC & EXTRN directives, it must be declared PUBLIC in the** | | |
| a) subroutine | | b) procedure |
| c) main routine | | d) main routine and subroutine |
| 1. **Which of the following statements is incorrect?** | | |
| a) complete code of instruction string is inserted at each place, wherever the macro name appears | | b) macro requires less time of execution than that of procedure |
| c) macro uses stack memory | | d) macro name can be anything except registers and mnemonics |
| 1. **Which of the following is not a mode of data transmission?** | | |
| a) simplex | | b) duplex |
| c) semi duplex | | d) half duplex |
| 1. **If the data transmission takes place in either direction, but at a time data may be transmitted only in one direction then, it is of** | | |
| a) simplex mode | | b) duplex mode |
| c) semi duplex mode | | d) half duplex mode |
| 1. **The 16-bit registers are available with their extended size of 32 bits, by adding the registers with a prefix of** | | |
| a) X | | b) E |
| c) 32 | | d) XX |
| 1. **Which of the following is a data segment register of 80386?** | | |
| a) ES | | b) FS |
| c) GS | | d) all of the mentioned |
| 1. **Match the following** | | |
| **a) DB** | **1) reserve only 10-bytes** | |
| **b) DT** | **2) reserve only 4 words** | |
| **c) DW** | **3)reserve byte or bytes** | |
| **d) DQ** | **4)reserve words** | |
| a) a-3, b-2, c-4, d-1 | | b) a-2, b-3, c-1, d-4 |
| c) a-3, b-1, c-2, d-4 | | d) a-3, b-1, c-4, d-2 |
| 1. **What will be the contents of register AL after the following has been executed**   **MOV BL, 8C**  **MOV AL, 7E**  **ADD AL, BL** | | |
| A) 0A and carry flag is set | | B) 0A and carry flag is reset |
| C) 6A and carry flag is set | | D) 6A and carry flag is reset |
| 1. **Which flag does the 80x86 use to check for unsigned arithmetic overflow?** | | |
| a) OF | | b) CF |
| c) SF | | d) None of above |
| 1. **If CS = 24F6H, SS = 0801H, SI = 0100H and IP = 634AH the address of the next instruction is:** | | |
| a) 25761 | | b) 08110 |
| c) 25060 | | d) 2B2AA |
| 1. **If ES = D321H, then the range of physical addresses for the extra segment is:** | | |
| a) 00000H – 0D321H | | b) D3210H – D321FH |
| c) D3210H – E320FH | | d) 0D321H - 1D320H |
| 1. **Assume AX=6521 H, BX=ABCD H and CX=0105 H. What are the flags after CMP AH, CL** | | |
| a) C=1, Z=0, S=1 | | b) C=0, Z=1, S=0 |
| c) C=0, Z=1, S=1 | | d) C=0, Z=0, S=0 |
| 1. **What is the result in AL after executing the following instructions?**   **XOR AL, AL**  **OR AL, 80H** | | |
| a) 80H | | b) 88H |
| c) 00H | | d) None of the above |
| 1. **Which of the following is an invalid instruction?** | | |
| a) add dx ,dx | | b) MOV AX, CS |
| c) sub bar,5 | | d) MOV AL, DI |
| 1. **Which directive(s) are used when defining 64-bit integers?** | | |
| a) DT | | b) DQ |
| c) DW | | d) None of the above |
| 1. **Which of the following is an invalid instruction?** | | |
| A) MOV AX, [BX+4] | | B) POP CH |
| C) XCHG AL, BYTEVAL[BX] | | D) ADD AX, [SI+3] |
| 1. **Assume AX=6521 H, BX=ABCD H and CX=0105 H. What are the contents of BL after OR BL, 1FH** | | |
| a) DF H | | b) FC H |
| c) D2 H | | d) DD H |
| 1. **What will be the value of the Overflow flag after executing the following instructions?** | | |
| **MOV AL, 80H** | |  |
| **ADD AL, 92H** | |  |
| a) OF = 0 | | b) OF = 1 |
| 1. **The value of AX after executing the following code is --------**   **MOV AX, 1461h**  **MOV DS, AX**  **MOV AL, 05**  **MOV AH, 07** | | |
| a) 1461h | | b) 0705 |
| c) 0507 | | d) 6114 |
| 1. **In the following data definition, assume that List2 begins at offset 2000h. What is the byte containing (5)?**   **List2 DW 3,4,5,6,7** | | |
| a) 2004H | | b) 2006H |
| c) 2005H | | d) 2003H |
| 1. **Real-address mode can be used to access how much memory?** | | |
| a) 1 megabyte | | b) 1 gigabyte |
| c) 2 megabytes | | d) 2 gigabytes |
| 1. **Which of the following defines a constant Max = 80?** | | |
| a) Max db 80 | | b) Max equ 80 |
| c) mov Max, 80 | | d) Max dw 80 |
| 1. **Which of the following, when used in the data section of a MASM program, reserves 40 bytes of RAM (memory)?** | | |
| a) DB 20 DUP (2) | | b) DW 40 DUP (1) |
| c) DB 20 DUP (20) | | d) DW 20 DUP (2) |

**Q2- [10] Allocate syntax errors and write V(Valid) or I (Invalid) for each of the following instructions**

|  |  |
| --- | --- |
| 1 | MOV DS, CS |
| 2 | MOV CS, SIZE |
| 3 | MOV [SI], [DI] |
| 4 | INC NUM3, 1 |
| 5 | MOV AX, [BX+4] |
| 6 | MOV SP, BP |
| 7 | XCHG AL, NUM2 |
| 8 | SUB DS, 3H |
| 9 | MOV BX, AH |
| 10 | SUB CHAR1, 'A' |

**Q3-[6] Write a single instruction for each of the following operations. Note that no other changes should occur.**

1. [1] Clear zero flag if AX is negative
2. [3] Three different instructions that will subtract 1 from register DX
3. [1] Set the 5th and the 11th bits in AX.
4. [1] Set zero flag if AX is even

**Q4- [4] What does the following programs do**

|  |  |
| --- | --- |
| **1** | **2** |
| MOV AH,0  MOV AL,13H  INT 10H  MOV CX,100  MOV AL,5  MOV AH,0CH  BACK: MOV DX,50  INT 10H  MOV DX,100  INT 10H  LOOP BACK  MOV DX,100  BACK2:MOV CX,100  INT 10H  MOV CX,0  INT 10H  DEC DX  CMP DX,50  JNZ BACK2 | .MODEL SMALL  .DATA  V DB 1,4,5,3,2  N DB ?  .CODE  MAIN PROC FAR  MOV AX,@DATA  MOV DS,AX  MOV DL,V[0]  MOV CX,OFFSET N  DEC CX  MOV BX,0  LBL: INC BX  MOVAL,V[BX]  CMPDL,AL  JB LBL2  MOVDL,AL  LBL2: LOOP LBL  MOV N,DL  HLT  MAIN ENDP  END MAIN |

**Q5- [Total: 60] Write the following programs [*DO NOT FORGET COMMENTS*]**

1. [10] Write an 8088/8086 assembly language subroutine which will count the number of times that a specified ASCII character occurs within a ZERO terminated string. The subroutine will be called with DS:SI pointing to the string to be searched, and AL containing the ASCII character to search for, and should return with the count in AX. The subroutine should make no assumptions about the state of any flags, and should return with all registers preserved (except AX, which contains the return value). The subroutine should be declared using the PROC directive.

Note: A zero terminated ASCII string is a sequence of ASCII character codes with the end of the sequence indicated by a byte containing the value 0. For example: If the subroutine were called with a pointer to the following string in DS:SI, and the value ‘m’ in AL, it would return 3 in AX:

STR DB “Programming in assembler is easy”,0

1. [10] X is an array that contains duplicate values. for example, X = [11222333447899]. Write a program that removes these duplicates. Therefore, the result in this case should be X= [1234789].
2. [10] Draw a line by reading length, starting point and angle from user. Possible angle values are [0,45,90,135,180]
3. [5] Using string operations, write a program that copies an array of words which offset is 0123h into a new array which has the offset 340Ah.
4. [5] Write a program that computes the sum of **M = 21+22+......+2N**, where N is entered by user.

*[HINT: N is less than or equal to 15]*

1. [5] Write a program that reads a number from user and check if it is divisible by 5 or not.
2. [5] Write a program to find the second occurrence location of a letter in a string.
3. [5] Write a program that converts a given string from upper case letters to lower case ones.
4. [5] Write a program that calculates the summation of all prime numbers till x where x is entered by the user.