COMSATS University Islamabad, Lahore Campus

Department of Computer Engineering

Microprocessor Systems and Interacting (CPE342)

Course Instructor: Engr. Usman Rafique

Assignment __TWO__ **Section: _FA22-BCE-B_**

Submitted by: Aliyan Ahmed Cheema

Reg. number: FA22-BCE-028

Submitted on: 27th March 2025

	Q1	Q2	Q3	Total
Marks Obtained				

COMSATS UNIVERSITY ISLAMABAD, Lahore Campus

Department of Computer Engineering

Subject: Microprocessor Systems and Interfacing (CPE342)		**Batch:**	**FA22-BCE-B**
Assignment No.	**TWO**	**Total Marks:**	**30**
Handed over on:	**20th March 2025**	**Submission Date:**	**27th March 2025 (In class)**
Student's Name:	Aliyan Ahmed Cheema		
Registration Number:	FA22-BCE-028		

Instructions:

- Provide your solution in the space provided against each problem
- Back side of each leaf is for rough work only
- Submission after the deadline will not be graded
- Do not use lead pencil in your solution

Note: The CPU referred to in this problem sheet is Intel 8086-88.

Problem 1 **10 Marks**

Consider the following high-level language program structure. Construct complete assembly language program that can be generated from this program structure.

```
for(a=0; a<=250; a=a+2){
     y = (x*a)-125;
          if(y>100){
               pair();
               y++;
          }    //end if
}    //end for

void pair(j,k){
     j = j/k;
}    //end void
```

Assume all the variables **a, x, y, j** and **k** are stored in BL, DL, DH, CL and CH, respectively.

Solution:

```
        JMP     START

; Implements: j = j/k   (i.e. CL = CL / CH)
PAIR:
        MOV     AL, CL      ; move j into AL
        DIV     CH          ; divide AL by k (CH), quotient in AL
        MOV     CL, AL      ; store result back in j
        RET

; Main Program
START:
        ; Initialize variables
        MOV     BL, 0       ; a = 0
        ;we are assuming that x,y,j,k are already in DL,DH,CL,CH

FOR_LOOP:
        CMP     BL, 250     ; compare a with 250
        JA      DONE        ; if a > 250, exit loop

        ; Compute y = (x * a) - 125
        MOV     AL, BL      ; AL = a
        MUL     DL          ; multiply: AL * x; result in AX (low 8-bit in AL)
        MOV     DH, AL      ; y = low byte of product
        SUB     DH, 125     ; y = y - 125

        ; If y > 100 then call PAIR and increment y
        CMP     DH, 100
        JLE     SKIP        ; if y <= 100, skip the if block

        CALL    PAIR        ; call subroutine pair(j,k)
        INC     DH          ; y++

SKIP:
        ADD     BL, 2       ; a = a + 2
        JMP     FOR_LOOP        ; repeat loop

DONE:
        HLT             ; halt program

        END
```

Problem 2 **10 Marks**

Construct an assembly language program that reads 1-byte data from 2000H and 1-byte data from 5000H. Place this data onto the stack, which is initially empty, such that the byte read from 1234H is the lower-byte and from ABCDH is the higher-byte of the 16-bit data that is to be placed onto the stack. Repeat this operation unless the stack segment becomes completely full. The stack segment starts from 9000H.

Solution:

```
START:
        ; Assuming there is already a memory in DS

        ; Set up the stack segment.
        MOV     AX, 9000H   ; stack segment starts at 9000H
        MOV     SS, AX
        ; Assuming SP is at some point in stack segment and stack is not empty

PUSH_LOOP:
        CMP     SP, 0
        JE      DONE            ; when SP becomes zero, stack is full
        ; Read the 1-byte lower and 1-byte higher parts.
        MOV     AL, [2000H] ; lower byte from memory location 2000H
        MOV     AH, [5000H] ; higher byte from memory location 5000H

        ; Push the word onto the stack.
        PUSH    AX

        JMP     PUSH_LOOP       ; repeat the operation

DONE:
        HLT             ; halt the program

        END
```

Problem 3 **10 Marks**

Construct an assembly language program that reads a word from stack segment and check whether it is odd or even. Check entire stack segment for this operation. Place the count of even words in AX and count of odd words in BX. The stack starts at B000H. Make use of a subroutine that checks the odd/even property of the word.

Solution:

```
        JMP    START       ; jump over the subroutine

    CHECK_ODD:
        TEST   DX, 0001h   ; test LSB of DX
        JNZ    IS_ODD      ; if nonzero, it is odd
        INC    AX          ; even: increment even counter in AX
        RET
    IS_ODD:
        INC    BX          ; odd: increment odd counter in BX
        RET


    START:
        ; Set stack segment to B000H
        MOV    AX, B000H
        MOV    SS, AX

        ; Initialize SP to 0
        MOV    SP, 0000H

        ; Initialize counters: even count in AX = 0, odd count in BX = 0
        MOV    AX, 0
        MOV    BX, 0

        ; Set CX = 32768 words (64KB/2) to scan the entire segment
        MOV    CX, 32768

    SCAN_LOOP:
        ; Read a word from the stack segment to DX.
        POP DX
        CALL   CHECK_ODD   ; update AX (even) or BX (odd)
        LOOP   SCAN_LOOP

        ; At this point, AX = even count, BX = odd count.

    HLT

        END
```