National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Computer Org. & Assembly Lang.	Course Code:	EE2003
Program:	BS (CS), BS (DS)	Semester:	Spring 23
Exam Duration:	60 mins	Total Marks:	30
Date:	11-4-2023	Weight	12.5%
Exam Type:	Mid II	Pages:	7

Name: **Roll No.:** **Section:**

Instructions:

1. Attempt all questions **on the answer booklet**.
2. If you think some information is missing then make an assumption and state it clearly.
3. This is an **open book and open notes** exam.
4. Sharing of notes or calculators is strictly prohibited.

Question 1 [CLO 2] **[2 + 2 + 6 marks]**

A. Consider a subroutine TempSBR that returns three output values on stack. How will you allocate space on stack for these output variables before calling TempSBR? Write a single statement for this task.

```
sub sp, 6
```

B. You want to display letter 'P' on the screen, row 5, column 3 in default white-on-black color. Complete the following code by writing the missing statement.
Note: row & column numbering starts at zero.

```
mov ax, 0xb800
mov es, ax
……
```

```
mov word [es:806], 0x0750
```

C. In the following code, work out the (hex) values in AX and DX registers after following lines have finished.

(i) Line 3 (ii) Line 7 (iii) Line 11

```
 1  mov BX, 650
 2  mov AX, 820
 3  mul BX

 4  mov DX, 0
 5  mov AX, 85
 6  mov BX, 8
 7  div BX

 8  mov DX, 2
 9  mov BX, 3
10  mov AX, 900
11  div BL
```

	DX	AX
(i)	08	2208
(ii)	05	0A
(iii)	02	division error

Question 2 [CLO 2] **[5 marks]**

Consider the following subroutine. It calculates the factorial of a number placed in stack as parameter. Factorial result is returned in the AX register (assume that output fits in 16 bits).

This code has some logical errors. Correct those errors so that the required functionality can be achieved. Do not add any new lines, just correct the existing ones.

```
factorial:
    push bp
    mov bp, sp
    mov ax, [bp+2]
    cmp ax, 0
    ja L1

    mov ax, 1
    jmp L2

L1: dec bp
    push bp
    call factorial

    mov bx, [bp+2]
    mul bx
L2: pop bp
    ret 4
```

```
factorial:
    push bp
    mov bp, sp
    mov ax, [bp+4]
    cmp ax, 0
    ja L1

    mov ax, 1
    jmp L2

L1: dec ax
    push ax
    call factorial

    mov bx, [bp+4]
    mul bx
L2: pop bp
    ret 2
```

Question 3 [CLO 3] [3 + 2 + 2 marks]

Dry run the code below to answer the following questions.

(a) What will be printed on screen by this program?
(b) The offset of `str1` is `0x103`. What will be the value in DI register before line # 14?
(c) Suppose lines 8-12 are deleted from this code. What will be the program output then?

```
 1  [org 100h]
 2  jmp start
 3
 4  str1: db 'mirror'
 5  str2: db 'home'
 6
 7  start:
 8      std
 9      mov si, str2+3
10      mov di, str1+5
11      mov cx, 3
12      rep movsb
13
14      mov ax, 0xb800
15      mov es, ax
16
17      mov ah, 0x07
18      mov si, str1
19      mov di, 0
20      mov cx, 6
21  repeat:
22      mov al, [si]
23      mov [es:di], ax
24      inc si
25      add di, 2
26      loop repeat
27
28      mov ax, 0x4c00
29      int 21h
```

Ans:

(a) mirome

(b) 0x105

(c) mirror

Question 4 [CLO 3] [8 marks]

Write a program to find how many times the last character is repeated at the end of a string. For example, given a string 'abc rte ebt dar keeee', the program should output 4 because the string ends with four occurrences of letter 'e'.

You **must** use the SCAS string instruction with appropriate repetition prefix to do the comparison and loop. Put the result in DX register.

Start the program with the following template

```
       [org 100h]
       jmp start

str1:  db 'abc rte ebt dar keeee'
len:   dw 21

start:
```

```
        [org 100h]
        jmp start

str1:  db 'abc rte ebt dar keeee'
len:   dw 21

start:
        std                 ; moving backwards
        mov di, str1
        add di, [len]
        dec di              ; point DI to last character (@ length-1)
        mov al, [di]        ; load last char in AL
        mov cx, [len]
        repe scasb

        mov dx, [len]
        sub dx, cx
        dec dx
finish:
        mov ax, 0x4c00
        int 21h
```

Marking

1 Direction flag
2 Initial DI pointer
1 Correct char in AL
1 Loop count in CX
1 Repe scasb
2 work out the correct count in dx

- End of Exam -