

Assignment 2 (Due Date: 9th November,2022)

1. Consider following code:

Give the contents of the following registers, along with the run-time stack, when the following instructions are executed. Initially, ESP = 00001FF8h.

```

X1 DWORD 25H, 27H
MAIN PROC
    PUSH 6
    PUSH 5
    CALL Addition
11500000H    MOV RESULT, EAX    ; ESP: _____
MAIN ENDP

Addition PROC

    PUSH OFFSET X1

    PUSH OFFSET [X1+4]    ; ESP: _____
    POP ESI
    ADD [ESI], EAX    ; [ESI]: _____

    MOV ECX, 5

    add eax, ecx

    ret

Addition ENDP
```

Draw a stack and write the output of each line where asked (this program may contain errors, first you need to deal with that error).

2. Consider following:

Mov al, 00100101b

Test al, 00001001b

Mov al, 00100100b

Test al, 00001001b

Write the status of ZF after execution of line 2 and 4

3. Create a procedure that saves the positive values of all array elements of an array1 in another array2. Write a test program that calls the procedure, passing a pointer to a signed doubleword array and the size of the array. Preserve all other register values between calls to the procedure.

Let elements of array1 be 40, -90, -67, 98, 78, -45, 0, 32.

4. Implement the following pseudocode in assembly language. Assume that A, B, and N are 32-bit signed integers.

while N > 0

if N != 3 AND (N < A OR N > B)

N = N - 2

else N = N - 1

end while

5. Write a program that asks the user to enter a number. If AL contains 1 or 3, display "o"; if AL contains 2 or 4, display "e".

6. Convert the following C code to assembly language.

```
int a = 100, b = 200, c,i,j;
    for ( i = 5; i >0; i--)
    {
        b = a + b;

printf ("%d", b);
    for (int j = 5; j > 0; j--){
        a = a -1;
        c+=10;
    }
    printf ("%d%d", a,c);
}
```

7. Write a program in assembly that takes an integer input from the user and display the following pattern. Suppose the input is 8 then print the following pattern of 8 line.

```
8 7 6 5 4 3 2 1
8 7 6 5 4 3 2
8 7 6 5 4 3
8 7 6 5 4
8 7 6 5
8 7 6
8 7
8
```

8. Transmitted messages often include a parity bit whose value is combined with a data byte to produce an even number of 1 bits. Suppose a message byte in the AL register contains 01110101. Show how you could use the Parity flag combined with an arithmetic instruction to determine if this message byte has even or odd parity.