Name: Muhammad Maiz Nadeem
Reg. ID: SP21-BCS-052

Question 1:

Write a program that calculates the first 12 terms of fibonacci sequence and stores them in an array.

Answer:

```
ORG 100h

.DATA

    ARRAY DW 12 DUP(?)

.CODE

        MOV CX, 12
        MOV SI, OFFSET ARRAY

        XOR AX, AX
        MOV BX, 0
        MOV DX, 1
        MOV [SI], BX
        ADD SI, 2
        MOV [SI], DX
        ADD SI, 2

FIBO:   XOR AX, AX
        ADD AX, BX
        ADD AX, DX
        MOV BX, DX
        MOV DX, AX
        MOV [SI], AX
        ADD SI, 2
        LOOP FIBO

RET
```

Question 2:

(Modified version of exercise from chapter 10)

To sort an array A of N elements by the bubblesort method, we proceed as follows:

- Pass 1: For j= 2 ... N, If A(j) <A(j - 1) then swap A(j) and A(j-1).
- This will place the largest element In position N.
- Pass 2: For j = 2 ... N -1, if A(j)< A(j-1) then swap A(j) and A(j -1).
- This will place the second largest element in position N -1.
- Pass N -1: If A(2) < A(1), then swap A[2] and A(1).

At this point the array is sorted.

Demonstration :

Initial Data:	7 5 3 9 1
Pass 1:	5 3 7 1 9
Pass 2:	3 5 1 7 9
Pass 3:	3 1 5 7 9
Pass 4:	1 3 5 7 9

Write a program segment to sort a byte array (DATA DB 7 5 3 9 1)by the bubblesort algorithm. The program should put the offset address of the array in SI and the number of elements in variable N.

Answer:

```
ORG 100h

.DATA

    SWAP DB ?
    N DW 6
    ARRAY DB 2, 6, 7, 1, 4, 7

.CODE

START:      MOV SWAP, 0
            MOV SI, OFFSET ARRAY    ; storing array offset in SI
            XOR BX, BX


COMPARISON: MOV AL, [SI+BX]
            CMP AL, [SI+BX+1]
            JBE NOSWAP

;SWAP:
            MOV SWAP, 1             ; indicates swapping has occurred
            XCHG AL, [SI+BX+1]
            MOV [SI+BX], AL


NOSWAP:     INC BX
            CMP BX, N               ; using N as number of elements to loop through array
            JNZ COMPARISON

            CMP SWAP, 0             ; checks if swapping has occurred
            JNZ START                   ; if yes, loops again. if not, program ends

RET
```

Question 3:

Implement the following sorting algorithm for a byte array:

- i = N
- FOR i=N-1 times DO
- Find the position k of the largest element among A[1] ….A[i]
- Swap A[k] and A[i]
- i=i-1
- END FOR

Usually algorithms are evaluated on speed by observing how many steps they took to sort a certain data set.

Answer:

```
ORG 100h

.DATA
    SWAP DB ?
    N DW 6
    ARRAY DB 2, 6, 7, 1, 4, 8

.CODE

MOV SI, OFFSET ARRAY            ; storing array offset in SI
SUB SI, 1
INC N                           ; retaining value of N for first cycle

START:

    DEC N                       ; decrementing N whenever the least number is caught
    MOV SWAP, 0                 ; indicates least value is caught
    INC SI                      ; incrementing SI whenever the least value is caught and stored in it
    XOR BX, BX


COMPARISON:
    MOV AL, [SI]
    MOV DL, [SI+BX+1]
    CMP AL, [SI+BX+1]
    JBE NEXT

;SWAP:
    MOV SWAP, 1
    XCHG AL, [SI+BX+1]
    MOV [SI], AL


NEXT:
    INC BX
    CMP BX, N                   ; using N as number of elements to loop through array
    JNZ COMPARISON

CMP SWAP, 0                     ; checks if least value is caught
```

```
JNZ START                       ; if yes, loops again. if not, program ends

RET
```

For the data set given in question 1 , which algorithm you feel is faster, Bubble sort or select sort? (Hint : See which algorithm does less swapping)

Selection sort requires less number of variables and has less cycles as it selects the indexes instead of exchanging them again and again.