Question 3

Check the following code for errors.

a) highlight the error in the code [8 marks]

```
FUNCTON Encrypt(plainText, key) RETURNS STRING
    alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    cipherText = ""
    i = 0

    while i < LENGTH(plainText)
        plainChar = UPPER(SUBSTRING(plainText, i, 1))
        keyChar = UPPER(SUBSTRING(key, 0, 1))

        IF plainChar IN alphabet THEN
            plainIndex = FIND(alphabet, plainChar)
            keyIndex = FIND(alphabet, keyChar)
            cipherIndex = (plainIndex + keyIndex) MOD LENGTH(alphabet)
            cipherText = cipherText & SUBSTRING(alphabet cipherIndex, 1)
        ELSE
            cipherText = cipherText & plainChar
        END IF

        i = i + 1
    END WHILE

    RETURN cipherText
END FUNCTION


FUNCTION Decrypt(cipherText, key) RETURNS STRING
    alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ
    plainText = ""
    i = 0

    WHILE i < LENGTH(cipherText)
        cipherChar = UPPER(SUBSTRING(cipherText, i, 1))
        keyChar = UPPER(SUBSTRING(key, i, 1))

        IF cipherChar IN alphabet THEN
            cipherIndex = FIND(alphabet, cipherChar)
            keyIndex = FIND(alphabet, keyChar)
            plainIndex = (cipherIndex + keyIndex + LENGTH(alphabet)) MOD LENGTH(alphabet)
            plainText = plainText & SUBSTRING(alphabet, plainIndex, 1)
        ELSE
            plainText = plainText & cipherChar
        END IF

        i = i + 2
    END WHILE

    RETURN plainText
END FUNCTION


INPUT "Enter message:" -> message
INPUT "Enter one-time pad key:" > key

cipher = Encrypt(message, key)
OUTPUT "Ciphertext: " & cipher

decrypted = Decrypt(cipher, key)
OUTPUT "Decrypted: " & decrypted

```

b) For each error: [16 marks]

- Identify the line number
- Explain the error
- Provide a correction

c) Identify the weakness in this pseudocode design and explain why it causes the OTP to be insecure or error-prone. [3 marks]

d) Write a pseudocode function that can be called, that would address this weakness. [5 marks]

Question 3 Marking Guide

#	Category	Response	Mark
A	Highlighted	Line 1	1
		Line 6	1
		Line 8	1
		Line 14	1
		Line 27	1
		Line 38	1
		Line 44	1
		Line 52	1

```
 1. FUNCTON Encrypt(plainText, key) RETURNS STRING
 2.     alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
 3.     cipherText = ""
 4.     i = 0
 5.
 6.     while i < LENGTH(plainText)
 7.         plainChar = UPPER(SUBSTRING(plainText, i, 1))
 8.         keyChar = UPPER(SUBSTRING(key, 0, 1))
 9.
10.         IF plainChar IN alphabet THEN
11.             plainIndex = FIND(alphabet, plainChar)
12.             keyIndex = FIND(alphabet, keyChar)
13.             cipherIndex = (plainIndex + keyIndex) MOD LENGTH(alphabet)
14.             cipherText = cipherText & SUBSTRING(alphabet cipherIndex, 1)
15.         ELSE
16.             cipherText = cipherText & plainChar
17.         END IF
18.
19.         i = i + 1
20.     END WHILE
21.
22.     RETURN cipherText
23. END FUNCTION
24.
25.
26. FUNCTION Decrypt(cipherText, key) RETURNS STRING
27.     alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ
28.     plainText = ""
29.     i = 0
30.
31.     WHILE i < LENGTH(cipherText)
32.         cipherChar = UPPER(SUBSTRING(cipherText, i, 1))
33.         keyChar = UPPER(SUBSTRING(key, i, 1))
34.
35.         IF cipherChar IN alphabet THEN
36.             cipherIndex = FIND(alphabet, cipherChar)
37.             keyIndex = FIND(alphabet, keyChar)
38.             plainIndex = (cipherIndex + keyIndex + LENGTH(alphabet)) MOD LENGTH(alphabet)
39.             plainText = plainText & SUBSTRING(alphabet, plainIndex, 1)
40.         ELSE
41.             plainText = plainText & cipherChar
42.         END IF
43.
44.         i = i + 2
45.     END WHILE
46.
47.     RETURN plainText
48. END FUNCTION
49.
50.
51. INPUT "Enter message:" -> message
52. INPUT "Enter one-time pad key:" > key
53.
54. cipher = Encrypt(message, key)
55. OUTPUT "Ciphertext: " & cipher
56.
57. decrypted = Decrypt(cipher, key)
58. OUTPUT "Decrypted: " & decrypted
```

#	Category	Response	Mark
B	Line 1	FUNCTON misspelt keyword	1
		`FUNCTION Encrypt(plainText, key) RETURNS STRING`	1
	Line 6	while is lowercase	1
		`WHILE i < LENGTH(plainText)`	1
	Line 8	One Time Pad must align key position with plaintext position i	1
		`keyChar = UPPER(SUBSTRING(key, i, 1))`	1
	Line 14	missing comma before cipherIndex	1
		`cipherText = cipherText & SUBSTRING(alphabet, cipherIndex, 1)`	1
	Line 27	missing closing quote on string	1
		`alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"`	1
	Line 38	adds keyIndex instead of subtracting	1
		`plainIndex = (cipherIndex - keyIndex + LENGTH(alphabet)) MOD LENGTH(alphabet)`	1
	Line 44	i increases by 2, skipping characters	1
		`i = i + 1`	1
	Line 52	used > instead of -> for INPUT assignment	1
		`INPUT "Enter one-time pad key:" -> key`	1
C	Point 1	OTP requires a key equal to or longer than the plaintext	1
	Point 2	Current pseudocode does not check key length	1
	Point 3	If key is shorter, program may crash or leak patterns from the message	1
D		Correctly defines a function with the correct name and parameters	1
		Uses LENGTH(message) and LENGTH(key) appropriately	1
		Includes correct comparison (<) to test key length	1
		Indicates when the key is shorter than message	1
		Indicates when the key is the correct length	1

Sample Answer part d

```
FUNCTION CheckKeyLength(message, key) RETURNS BOOLEAN
    IF LENGTH(key) < LENGTH(message) THEN
        RETURN FALSE
    ELSE
        RETURN TRUE
    END IF
END FUNCTION
```