

1) The following code is riddled with bugs: some are due to bad syntax and some are due to bad logic. There are 7 bugs, each bug is on its own line and each bug is worth one mark. Circle up to 7 lines; **one mark will be deducted for each additional line circled**. This question is scored out of 6 and a maximum of 7 will be awarded.

```
import random
MIN_ASCII_VALUE = ord('A')
MAX_ASCII_VALUE = ord('z')

def encode(strToEncode) :
    encodeList = tuple(strToEncode)
    randomList = [''] * len(strToEncode)
    encodeString = 'something'

    i = 0
    while i < len(encodeList)
        if (encodeList[i] == 'A') :
            # Switch As to Zs
            encodeList[i] = 'Z'
        elif (encodeList[i] = 'Z') :
            # Switch Zs to As
            encodeList[i] = 'A'

        # Create a random character for each original character
        randomList[i] = chr(random.randint(MIN_ASCII_VALUE, \
                                           MAX_ASCII_VALUE))

        i = i + 1

    # Reverse the list of original characters
    reversed(encodeList)

    # Create the encoded string
    for aChar, randChar, newChar in zip(encodeList, randomList) :
        # Add the encoded character followed by a random character
        encodeString = encodeString + aChar
        encodeString.append(randChar)

    # Return the string
    return encodeString
```

2) Examine the following small Python programs. Determine the values of the variables at the end of each program's execution. There are 11 variables and each is worth one mark. This question is scored out of 10; a maximum of 11 will be awarded.

```
a = 4
b = 5
c = 6
if (a > b) :
    c = 7
else :
    a = 8
    if (a < c) :
        b = 9
    else :
        c = 10
```

a: 8

b: 5

c: 10

```
x = 5 + 4 > 9 or 3 == 11 - 8
y = 4 <= 2 * 2 and 7 - 6 != 1
z = not "ABC" < "aBC"
```

x: True

y: False

z: False

```
i = 10
j = 0
while (i > 0) :
    i = i - 1
    break
else :
    j = 5
```

i: 9

j: 0

```
aList = [8, 4, 2, 6]
subList = aList[ : 2]
del aList[1]
test = 8 not in aList
```

aList: [8, 2, 6]

subList: [8, 4]

test: False

3. a) Write a function `chooseNumber(...)` that has two parameters, each representing a non-zero integer. The function must ask the user to choose between the two numbers and enter his or her choice. If the user does not enter one of the numbers then the function must ask again. Once the user has selected one of the numbers, the function must return the value entered. The following is sample output from calling `chooseNumber(5, 7)`:

```
You must choose between the numbers 5 and 7.  
Enter it now: 3  
You must choose between the numbers 5 and 7.  
Enter it now: 8  
You must choose between the numbers 5 and 7.  
Enter it now: 6  
You must choose between the numbers 5 and 7.  
Enter it now: 7
```

There is no need to write a `main()` function or a call to `chooseNumber(...)`. This question is scored out of 4.

```
def chooseNumber(num1, num2) :  
    choice = 0  
    while (choice != num1) and (choice != num2) :  
        print("You must choose between the numbers", num1, \  
              "and", str(num2) + ".")  
        choice = input("Enter it now: ")  
        choice = int(choice)  
  
    return choice
```

b) Write a function `printMixedString(...)` with two parameters, each representing strings. These strings must be the same size; if not, the function must print an error message. Each pair of characters in the parameter strings must be examined in order. One of them is selected for printing if:

- it is a lowercase 'j'
- its ASCII value is less than the other

The following is sample output from calling `printMixedString("Join", "ajar")` followed by `printMixedString("Hello", "ajar")`:

Jjan

You can't mix those!

There is no need to write a `main()` function or a call to `printMixedString(...)`. This question is scored out of 4.

```
def printMixedString(string1, string2) :
    if len(string1) != len(string2) :
        print("You can't mix those!")
        return

    string3 = ''
    for char1, char2 in zip(string1, string2) :
        if char1 == 'j' or char2 == 'j' :
            string3 = string3 + 'j'
        elif char1 < char2 :
            string3 = string3 + char1
        else :
            string3 = string3 + char2

    print(string3)
```