

Assignment 4

Part - I (5 points)

This part of the assignment contains multiple subparts. Make sure to complete all subparts to get full credit. You must use (nested) loops to program the solution to problems mentioned below. You are not allowed to use print statements that print multiple numbers at the same time. For example: to print numbers from 0 to 9 in the same line, you are not allowed to use:

```
print(0,1,2,3,4,5,6,7,8,9)   or   print("0 1 2 3 4 5 6 7 8 9")
```

Instead you should use:

```
for i in range(10):
```

```
    print(i, end = " ")
```

1) Part a: Write code that will print the following:

```
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
```

HINT: You need to use two for loops (one nested inside the other). First, create a loop that prints the first line (DO NOT PRINT USING `print("0 1 2 3 4 5 6 7 8 9")`). Then enclose it in another loop that repeats the line 10 times. Use either `i` or `j` variables for what the program prints. This problem and the next one helps reinforce what those index variables are doing.

Part b: Adjust the prior program to print:

```
0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9
```

2) Write code that will print the following:

```
0
0 1
0 1 2
0 1 2 3
0 1 2 3 4
0 1 2 3 4 5
0 1 2 3 4 5 6
0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7 8
0 1 2 3 4 5 6 7 8 9
```

HINT: This is same as problem (1), but the inside loop no longer loops a fixed number of times. Don't use `range(10)`, but adjust that range amount.

Extra Credit (2 points):

Write a Python program that will print the following:

```
10
11 12
13 14 15
16 17 18 19
20 21 22 23 24
25 26 27 28 29 30
31 32 33 34 35 36 37
38 39 40 41 42 43 44 45
46 47 48 49 50 51 52 53 54
```

- Generate the output using two `for` loops, one nested.
- Create a separate variable to store numbers that will be printed.

This problem requires a bit of an “a-ha” to get. Make sure to ask around if you have problems.

Part - II (5 points)

Decoding a secret message:

The description may seem daunting, but the solution is not that hard. You can use the built-in string datatype with the associated built-in functions and while loop (with ‘len’ function) or a for loop (with ‘in’ operator) to traverse the string. Also, use the ‘chr’ and ‘ord’ functions (which are based on ASCII code) discussed in course material. Make sure to look at the examples in the course material and do #18 and #19 in Exercises 2. Answer for #19 is provided and it can give valuable hints for solving this problem.

Your country is at war and your enemies are using a secret code to communicate with each other. You have managed to intercept a message that read as follows:

:mmZ\dxZmxjZpgy

The message is obviously encrypted using the enemy's secret code. You have just learned that their encryption method is based upon the ASCII code (you can find this set easily by searching online). Individual characters in a string are encoded using this system. For example, the letter 'A' is encoded using the number 65 and 'B' is encoded using the number 66.

Your enemy's secret code takes each letter of the message and encrypts it as follows (using a secret key):

If $(\text{OriginalChar} + \text{Key} > 126)$ then

$\text{EncryptedChar} = ((\text{OriginalChar} + \text{Key}) - 127) + 32$

Else

$\text{EncryptedChar} = (\text{OriginalChar} + \text{Key})$

For example, if the enemy uses $\text{Key} = 10$ then the message "Hey" would be encrypted as:

Character ASCII

H 72

e 101

y 121

Encrypted H = $(72 + 10) = 82 = \text{R}$ in ASCII

Encrypted e = $(101 + 10) = 111 = \text{o}$ in ASCII

Encrypted y = $32 + ((121 + 10) - 127) = 36 = \$$ in ASCII

Consequently, "Hey" would be transmitted as "Ro\$".

Write a program that decrypts the intercepted message. You only know that the key used is a number between 1 and 100. Your program should try to decode the message using all possible keys between 1 and 100. When you try the valid key, the message will make sense. For all other keys, the message will appear as gibberish.

HINT: You will need to implement a decrypt function that takes in an encrypted message as string and a key as integer and returns the decrypted message as string. You can decrypt each letter of the message as follows:

If $(\text{EncryptedChar} - \text{Key} < 32)$ then

$\text{DecryptedChar} = ((\text{EncryptedChar} - \text{Key}) + 127) - 32$

Else

$\text{DecryptedChar} = (\text{EncryptedChar} - \text{Key})$

NOTE: You should also implement an encrypt function that takes in a regular message as string and a key as integer and returns the corresponding encrypted message as string (the algorithm to encrypt a message is mentioned above in the problem description). This function would help you in encrypting any regular message, which then can be passed to your decrypt function to be decrypted.

For Encryption: You should ask the user for any regular message and a key and output the corresponding encrypted message.

Sample run:

Enter a regular message to encode:

Attack at dawn!

Enter a key value (between 0 and 100) for encoding:

88

The encoded message is:

:mmZ\dxZmx]Zpgy

For Decryption: You should ask the user for an encrypted message and output 100 well-formatted, decrypted messages (using keys between 1 and 100) along with the corresponding key value.

Sample run (the gibberish messages below are not accurate):

Enter an encrypted message to decode:

:mmZ\dxZmx]Zpgy

The following are the decoded messages for keys 1 to 100:

Key: 1 → Decoded Message: whfuihwuiidh89

Key: 2 → Decoded Message: 9ehkaOY3ewine

...

...

Key: 87 → Decoded Message: Buubdl!!bulebxo"

Key: 88 → Decoded Message: Attack at dawn!

...

Key: 100 → Decoded Message: on3dwp389/wi8