



CMPT 103 – Lab #3

General Information

Python version and IDE:	Python 3 / Wing IDE 101
Allocated lab time:	2 hrs and 50 min
Due date:	At the end of the lab period
Lab weight:	3%

Topics

- ✓ Text and Mutability

Submission

- ✓ **All the code files (.py) should be submitted electronically** to your Lab Blackboard site.
- ✓ A portion of the total marks (20%) will be allocated for the programming style. For example, functions should be small; avoid writing duplicate code; names should be meaningful and descriptive; naming convention should be followed consistently; code should be formatted properly; and comments should be accurate and well written.
- ✓ Comments are **required** for:
 - EACH program indicating the student name and program name.
 - EACH function indicating the function purpose, syntax (example usage of the function), parameters, and return value.
 - Any block of code for which the purpose may be unclear (Note: you should always try to write clean code that can be understood easily without comments).

Assignment

For this lab, please put all functions into a file called `Lab3your_initials.py` (e.g., `Lab3FL.py` where F and L are the first letter of your first name and last name). Please feel free to write helper functions if necessary.

1) Simple encryption and decryption:

- [25 marks] Create a function named `encrypt` that accepts a string as an argument and that returns an encrypted string. The `encrypt` function should replace each character in the string with a new character. The ASCII value of the new character is obtained by adding $(x + 5)$ to the original ASCII value, where x is the index (location) of the character in the encrypted string.

You can use the built-in function `ord` to obtain a character's ASCII value and the built-in function `chr` to obtain the character for an ASCII value.

For example: the word "Hello" will be encrypted as "Mkstx". The steps to encrypt it are as follows:

Character	ASCII	Location	New character's ASCII code	Encrypted character
H	72	0	77 (72+0+5)	M
e	101	1	107 (101+1+5)	k
l	108	2	115 (108+2+5)	s
l	108	3	116 (108+3+5)	t
o	111	4	120 (111+4+5)	x

```
>>> result = encrypt('Hello')
>>> print(result)
Mkstx
```

- b. [25 marks] Create a function named `decrypt` that accepts a string as an argument and that reverses the action of `encrypt`.

```
>>> original_str = decrypt('Mkstx')
>>> print(original_str)
Hello
```

2) Compression and expansion functions for strings:

- a. [25 marks] Write a function called `compress` that compresses a string by replacing any repeated letters with a letter and number. The function should return the shortened version of the string.

```
>>> print(compress(' '))

>>> print(compress('abc'))
abc

>>> print(compress('aaabbbbbbbbbbcccccc'))
a3b10c7
```

- b. [25 marks] Write a function called `expand` that takes a string in the compressed format and that returns the original string. You may assume that any number in a compressed string is a single-digit number.

```
>>> print(expand(' '))

>>> print(expand('abc'))
abc

>>> print(expand('a3bc5'))
aaabccccc
```

Optional bonus question (10 bonus marks):

Modify your `expand` function so that it can expand any compressed strings, including those that contain more than single-digit numbers. Please name your function `expand_any`.

```
>>> print(expand_any('a12b10cd5'))  
aaaaaaaaaaaabbbbbbbbbcdddd
```

Note: *No assignment will exceed the maximum of 100/100 even with the bonus marks.*