

# 1.2 - Week 1 - Applied - Practical

---

## Overview

These exercises' objectives are for you to:

- Get accustomed to our expectations for all the assessments, including commenting code and coding style.
- Become familiar (perhaps again) with Python



**NOTE:** Do not use Python functions that would make these tasks trivial (for example - count function for task 1). Do not use index splicing or negative indexing to accomplish these tasks. **Although these exercises are unmarked, they are here for you learn what will be expected of your solutions in the assignments.**



### IMPORTANT

Before attempting these exercises, read example documentation and further explanation of good documentation under Important Documents on Moodle. This will ensure you know how you are expected to comment Python code in this unit.

You will see two buttons in your task console:



**RUN** - The run button will run your python file in the scaffold. You can put some extra code in the main section of your code if you want to run that code to test your logic



**MARK** - This button will run the test cases that have been added to check your code. They are here for you to test your solution for correctness. **Note that these tests are not exhaustive and passing all these tests doesn't ensure complete correctness of your solution.**

---

# IMPORTANT

Since we are going to be using certain tools this semester, please go through the steps to set up -

1. PyCharm
2. MARS Simulator
3. Github Desktop / GitLab

Your demonstrators will help you set these tools up. Please make sure these are done before you finish the lab as these are extremely important.

---

## Task 1

Later in the unit, you might want to use a menu to access their functionality. The code in `Task1.py` gives you an idea of how to start a simple menu for reading items into a list, printing the list, and reversing it. Later in the unit, we will see much better ways of doing this but, for now, this is simple enough.

Add the above code to the `Task1.py` file, and extend it so that a user can perform the following commands on `my_list` using the menu:

- `last`, which returns a new list with the last item removed, after printing the item.
- `count`, which returns the number of times a given value appears in the list.

Please see the [appendix](#) for sample output.

---

## Task 2

Implement a program in file `Task2.py` that provides a menu to convert positive integers from decimal representation into hexadecimal and binary. Your menu should use 1 for converting a number in decimal representation to binary, 2 for converting to hexadecimal, and 3 for quitting. For options 1 and 2 it should also print the result. As usual, make sure you use good code layout, with consistent and meaningful variable names, well commented, and you modularise if appropriate.



**NOTE:** Do not to use any external libraries for this task.



### Information:

The algorithm to convert an integer **n** to binary has the following steps:

- divide **n** by 2 (integer division)
- take the remainder
- repeat the above two steps until the result of the division is 0.

The binary value is then the sequence of remainders from last to first.

See this [appendix](#) for an example and some sample output

---

## Task 3

For Task 3, Add to `Task3.py` a function that converts any decimal number into any base **(to make it easier, let's say it is a base from 2 to 16)** and returns the resulting string, where both the base and decimal number are provided as input.

No menu is required for this task



### Information:

The algorithm to transform a decimal number to octal (base 8) is the same as for hexadecimal (base 16) and binary (base 2). The only difference is the base used for the division (8 for octal, 16 for hexadecimal, and 2 for binary), and the fact symbols other than digits need to be used when the remainder exceeds 9 (that is, the base is 10 or higher, as is the case for hexadecimal).

See this [appendix](#) for an example and some sample output

---

## Appendix 1

```
Menu
1. append
2. reverse
3. print
4. last
5. count
6. quit

Enter command: 1
Item? 8

Menu...
Enter command: 1
Item? 3

Menu...
Enter command: 1
Item? 8

Menu...
Enter command: 1
Item? 5

Menu...
Enter command: 3
['8', '3', '8', '5']

Menu...
Enter command: 2

Menu...
Enter command: 3
['5', '8', '3', '8']

Menu...
Enter command: 5
Item? 1
0

Menu...
Enter command: 5
Item? 3
1

Menu...
Enter command: 5
Item? 8
2

Menu...
Enter command: 4
8
```

(The menu has not been printed multiple times to save scrolling but you can choose to display it or not each time)

Menu

1. append

2. reverse

3. print

4. last

5. count

6. quit

Enter command: 1

Item? 8

Menu...

Enter command: 1

Item? 3

Menu...

Enter command: 1

Item? 8

Menu...

Enter command: 1

Item? 5

Menu...

Enter command: 3

['8', '3', '8', '5']

Menu...

Enter command: 2

Menu...

Enter command: 3

['5', '8', '3', '8']

Menu...

Enter command: 5

Item? 1

0

Menu...

Enter command: 5

Item? 3

1

Menu...

Enter command: 5

Item? 8

2

Menu...

Enter command: 4

8



---

## Appendix 2

### Example of a decimal number (18) to binary:

first divide 18 by 2, obtaining 9 with remainder 0;  
then divide 9 by 2, obtaining 4 with remainder 1;  
then divide 4 by 2, obtaining 2 with remainder 0;  
then divide 2 by 2, obtaining 1 with remainder 0;  
then divide 1 by 2, obtaining 0 with remainder 1.

Since we obtained 0, the loop finishes, returning the string "10010" as the binary representation of decimal number 18.

### Sample output:

```
Menu
1. Decimal to Binary
2. Decimal to Hexadecimal
3. Quit

Enter command: 1
Enter number to be converted 2
10

Menu...

Enter command: 1
Enter number to be converted 15
1111

Menu...

Enter command: 2
Enter number to be converted 16
10

Menu...

Enter command: 2
Enter number to be converted 1234
4D2

Menu...

Enter command: 3
Maria-new:Week1 mbanda$
```

1. Decimal to Binary
2. Decimal to Hexadecimal
3. Quit

Enter command: 1

Enter number to be converted: 2

10

Menu...

Enter command: 1

Enter number to be converted: 15

1111

Enter command: 2

Enter number to be converted: 16

10

Enter command: 2

Enter number to be converted: 1234

4D2

Enter command: 3

---

## Appendix 3

### Sample output

```
Menu
1. Convert Decimal to Other Base
2. Quit

Enter command: 1
Enter number to be converted 1234
Enter base to be converted to 2
10011010010

Menu...

Enter command: 1
Enter number to be converted 1234
Enter base to be converted to 8
2322

Menu...

Enter command: 1
Enter number to be converted 1234
Enter base to be converted to 16
4D2

Menu...

Enter command: 2
Maria-new:Week1 mbanda$
```

(Menu is not required for this task)

Number 1234

Base 2

Output 10011010010

Number 1234

Base 8

Output 2322

Number 1234

Base 16

