Bonus Assignments I

1 Getting Started

This is a short recap what you have learned the previous days.

1.1 Connecting to UPPMAX

Open a terminal and start a ssh connection to UPPMAX as you learned on Monday.

```
$ ssh -X <username>@rackham.uppmax.uu.se
```

Note: `<username>` is replaced by your own UPPMAX username.

1.2 Create new directory

Before you start with the following assignments, please create a directory `PythonLab1` in your home directory (`~/`). In this directory save all the files you create during this assignment.

2 Bonus Assignment - Sequences

In this exercise we write a short Python program (named `<program_name>.py`, think of a reasonable program name and name your file accordingly. Replace `<program_name>` with your new program name).

Chose two variables, e.g. `A` and `B` and assign the sequences `GATTACA` and `TACCATAC` to these variables. Make sure that the two sequences are assigned as strings to their variables `A` and `B`. Then print these sequences.
Save everything you wrote and close the editor. Then you can run your program: `python3 <program_name>.py`

Then extend your program:

1. Concatenate both sequences in both ways (`AB` and `BA`) and print both options.
2. Print prefixes and suffixes of length 3 of both sequences `A` and `B`. Use the built-in function `len()` for determining the suffixes.
3. Print out the second sequence from the last to the first position (last position first, first position last).
4. Assign this inverted sequence to a third variable, you could use the variable name `C`, and print the value of this variable.
5. Print out the middle base of each sequence. When a sequence has an even number of bases, print out the base at the right position of the middle. Use the built-in function `len()` for this task.
   For example: For `A = "GTCA"` the program should print out `C`.

   Hint: There exist built-in functions to convert a number to an integer.
6. Count how often each base occurs in the first sequence (How often does `G` occur in the first sequence, then `A`, so on.) and print out this number for each base.
7. Count how often does `TA` occur in the second sequence and print out this number.

3 Bonus Assignment - Calculate the product of two numbers

Write in an editor the program `product.py` as introduced in the lecture, which calculates the product of two numbers `456` and `15`. Save the program code as `product.py` and run the program as described in the previous assignment.

1. Now calculate the product `234` and `24` additionally to the first product and print out both products (results) in one single line.

2. Change the program so that all numbers `456`, `15`, `234`, and `24` are saved in one list, called `l`. Change the print statement so that each number gets printed and also the product of the first two and the last two numbers.

4 Bonus Assignment - More sequences

Write in an editor a program, which has three lists `l`, `m`, and `n`. Each list contains several sequences. Save and run the program as described previously.

```
l: AGGTC, GATC, CTGCA, ATTCGT, ATGGT, GATC
m: CTGCA, GATC
n: CUAGCUA, GTATGG, GUAUC, GTAG
```

Note: Remember to store all sequences as strings in each of the lists.

Extend your program so that it can perform the following tasks.

1. Print each sequence in list `l`.
2. Print the first and last sequence in list `l`.
3. For each sequence in list `l` store the second position of each sequence in a new variable and print this new sequence.
4. Add this new sequence to the list `l`.
5. How long is list `l` now? Print out the length of list `l`.
6. Delete the second sequence of list `l`. Print list `l` and its length.
7. Divide the new list `l` into two equal parts and store the first half in a new list `l1` and the second half in a new list `l2`. Print both lists.
8. Concatenate list `l2` and list `l1` (in this order) and store it in a new list `l3`.
9. Remove all sequences in list `l`, which are also present in list `m`.
10. Invert all sequences in new list `l` and store them in a new list `l4`.
11. In list `n` a few RNA sequences (`U` instead of `T`) are present.

    - Change these sequences back to DNA sequences.
    - Delete the RNA sequences in list `n`.
    - Add the new DNA sequences at the same position of list `n`.

    When you print list `n` it should contain the following sequences in this order: `CTAGCTA`, `GTATGG`, `GTATC`, and `GTAG`.