Introductory exercises
Programming for Bioinformatics

These exercises have been created by Giuseppe Profiti

python-001

1. assign the value 7 to a variable, print it
2. assign the value 3.14 to another variable
3. check if the first variable value is even; what can you use?
4. evaluate the average of the two variables
5. evaluate the distance of the two variables from the average
6. Evaluate the average of the two distances
7. change the value of the first variable with its double, subtract 1 to it and put the result into a third variable
8. repeat all the steps from 2 to 5 using the new value of the first variable
9. assign values to the four variables x1, y1, x2, y2
10. Evaluate the [euclidean distance](#) for the points (x1,y1) and (x2,y2)
11. Assign a probability value to a variable (probability ranges from 0 to 1), evaluate the [information content](#) of the event with that probability
12. Repeat points 10 and 11 for more values

python-002 strings

1. assign the string "fire and ice" to a variable
2. print the third character in the string
3. print the fifth character
4. print the tenth character, print the last character, print the second to last
5. print the characters in even position
6. print the characters in odd position
7. print the first half of the string
8. print the whole string in reverse
9. count the number of "i" and "e" in the string
10. change the word "and" with "&"
11. check if the string contains the string "fire"
12. check if the string contains the string "re and"
13. check if the string contains the string "re &"
14. print the position of the first "e"
15. print the position of the last "e"
16. Change the string to something you like, repeat every stop from 2 to 9 and step 15
17. assign the string "234 4329 7654 8923" to a variable

18. find a way to print each value in the string increased by 3: i.e. from "2 3 4" print 5 then 6 then 7

python_003_functions

1. define a function called "increase" that adds one to a given number
2. define a function called "add" that returns the sum of two numbers
3. define a function called "add3" that returns the sum of three numbers
4. define a function called "add5" that returns the sum of five numbers
5. define a function that takes as input a number and a string and returns as many copies of the string as specified by the number. I.e. manyTimes(3,'dad') should return 'daddaddad'
6. modify the previous function so that it returns something like 'dad,dad,dad'
7. modify the function at point 5 so you can specify a separator instead of just using the comma like in point 6

python-004 lists and for loop

1. merge the lists [4,8,-9,"the"] and ["silent force", 4.67, 9]
2. repeat steps 11 and 12 of the **python-002 string** section using a list: i.e. "2 3 4" -> [5,6,7]
3. extract the numbers from the string "23|64|354|-123"
4. extract the positive numbers from the string "-1-987-6823-8261"
5. create a string from the list [3.14,6.333,98.12,23.1]
6. define a function called "addlist" that returns the sum of the elements of any list

python 006 if and files

1. Create a list of numbers from 3 to 12
2. print the numbers of the list that are odd
3. print the numbers of the list that are multiples of 5
4. change the list so it contains the numbers from 8 to 23
5. repeat steps from 9 to 10 for the new list

6. put the values 5,2,7,8,1,-3 in a list, in this order
7. print the first and the third value in the list
8. print the double of all the values in the list
9. print each value in the list after doubling it, subtracting 2 and dividing by 3
10. print the sum of all the numbers in the list
11. print the minimum value in the list
12. print the maximum value in the list
13. print the average value

14. put the string "avocado" in a variable and the string "radar" in another one

15. print both strings in reverse
16. evaluate which one of the two strings is palindrome (i.e. a string is palindrome if it can be read in the same way both from left to right and from right to left)
17. print the first half of the first string and the second half of the second one

18. ask the user for two strings
19. check if one string is the complement of the other (i.e. "AC" and "TG" -> yes)
20. print 3 rows: the first and the third for the sequences, the second one should contain "|" if the bases are complementary, "X" if not. Example:

```
ATTCGT
||X|X|
TAGGAA
```

21. rewrite the program from steps 18 to 20 so that it works for reverse complement (i.e. the user types the strings "ATTCGT" and "AAGGAT" and she gets the previous example as result)
22. Modify the program from point 20 or 21 in order to print also the number of non complementary positions (in the given example: 2)
23. Modify the previous programs in order to read the sequences from file, using them in pairs (i.e. first row with second row, third with fourth and so on) Note: each pair of strings must have the same length (i.e. first and second row must have the same length, for example 6, third and fourth row may have both length 9 etc)
24. Modify the previous programs in order to read the sequences from file, checking for each one with all the others of the same length for the one with the smallest number of non complementary positions. Print each string with its best match and score as in point 22. Example:

Input file

```
ATTCGT
TAGGAA
TCAGCA
AAAAAAAAA
TTTTTTTTT
```

Result:

```
ATTCGT and TCAGCA with 1 non complementary position
TAGGAA and ATTCGT with 2 non complementary positions
TCAGCA and ATTCGT with 1 non complementary position
AAAAAAAAA and TTTTTTTTT
TTTTTTTTT and AAAAAAAAA
```

25. Choose a number, then create a list of lists of numbers, each one with one more element than the previous. Each list should start from the second to last element of the previous list. Example, if the number is 4 and the first value is 1 you should get:

[[1], [0,1], [0,1,2], [1,2,3,4]]
26. Is the previous one working with any number of lists? Try different values.

What should you change in the code in order to read the list in step 6 from a file?

Check your understanding of Python
The following text contains examples of erroneous python code, you should be able to understand why it is wrong. You can try cut & pasting the code to Python IDLE to see error messages and results.

```
1=a
```

```
print a=1
```

```
print if a=1
```

```
print if a==1
```

```
if a=1:
    print a
```

```
a = 12
if 3 in a:
    print "a is bigger than 3"
```

```
a = 10
for i in a:
    print i
```

```
name = John
address = 1 rue de la gare
```

```
#I want to print the names of the days
days = [1,2,3,4]
words = ['monday', 'friday', 'saturday', 'sunday']
for day in days:
    print day
```

```python
#I want the total sum of a list of numbers
numbers = [1,2,3,4,10,15,27]
i = 0
for i in numbers:
     total = i
```

```python
#I want to print the second element of the list
l = ['dog','cat','horse']
print l['cat']
print l[2]
```

```python
#I want to print the list if 'cat' is in the second position
l = ['dog','cat','horse']
if l['cat']==2:
     print l
if l[2]==2:
    print l
```

```python
#I want to print all the grades
#and then the names of students with a good mark
grades = {'student1':27, 'student2': 18, 'student3': 'fail'}
print grades[]
print grades['students']
for zzz in grades:
     if zzz > 25:
            print zzz,"is a good student"
```

```python
#What is going on here?
def b(a,c):
    return a*c+2

def a(b,c):
    c = c + b(b,c)
    b = b(b,c)*c

def c(a,q):
    a = b(a,q)
    c = a+2
    return c
```