

# Repeat Exam BIO134: Programming in Biology HS2019

25.08.2020; 14:30-17:00

## Question 1

```
numbers = [5, -2, -3, 0, 4, 1, -7, 0, 8]
```

Write a program that creates two lists based on the list *numbers*. The first new list should contain all the positive numbers, the second list all the negative numbers in the order they occur in *numbers*; the zeros should be ignored.

The program should then print both new lists, first the list containing the positive numbers then the list with the negative numbers.

```
[5, 4, 1, 8]
[-2, -3, -7]
```

The program should still work according to the same principle if *numbers* was replaced by another list of different length and content.

## Question 2

Here is a list of UZH rectors (presidents) in chronological order:

```
rectors = [['Weder', 'Fischer', 'Jarren', 'Hengartner'],
           ['Hans', 'Andreas', 'Ottfried', 'Michael'],
           [2000, 2008, 2013, 2014]]
new_rector = ['Schaepmann', 'Michael', 2020]
```

Write a program that adds the new rector from the list *new\_rector* to the list *rectors* and prints the updated list. Thus, the program should print:

```
[['Weder', 'Fischer', 'Jarren', 'Hengartner', 'Schaepmann'],
 ['Hans', 'Andreas', 'Ottfried', 'Michael', 'Michael'], [2000, 2008,
 2013, 2014, 2020]]
```

The program should still work if the list *rectors* contained information about a different number of past rectors.

### Question 3

Write a function *affordable()* that takes a non-empty list of numbers plus a single number as input arguments. The function should test whether the sum of the numbers in the list is not larger (but rather smaller or equal) than the single number.

Call the function using the following code:

```
groceries = [2.50, 5.95, 0.6, 19.95, 3.20, 1.50]
limit = 30

if affordable(groceries, limit):
    print('You can afford it!')
else:
    print('Sorry, too expensive...')
```

You are not allowed to use *sum()* or any other related function that sums up numbers in a list.

### Question 4

The file *microbe\_identifiers.txt* contains microbial identifiers separated by spaces.

```
S97-cy101 S97-cy10339 S97-ga15914 S97-fi20693 ...
```

Each identifier starts with the prefix *S97-* and is followed by a taxon ID that consists of two letters and an integer number, eg *cy101* for the first microbe in the file.

Write a program that opens the file, extracts all the identifiers and creates a dictionary to store the taxon IDs without the prefix. The dictionary should use the two letters of the taxon ID as keys. The values should be lists containing all the taxon IDs that start with the respective two letters in the order they occur in the original file.

It should then print this dictionary:

```
{'cy': ['cy101', 'cy10339', 'cy57'], 'ga': ['ga15914', 'ga2201'],
'fi': ['fi20693', 'fi7767'], 'de': ['de32927', 'de522', 'de7777'],
'ac': ['ac375', 'ac47381', 'ac517', 'ac8603']}
```

Note that a dictionary does not have a specified order, so that the same dictionary can be printed in many different ways.

Your program should still work if the file contained a different number of other microbial identifiers.

## Question 5

```
import numpy as np
im = np.array(
[[[0.7, 0.4, 0.2],[0.5, 0.8, 0.8],[0.1, 0.9, 0.2],[0.3, 0.4, 0.5]],
[[0.7, 0.3, 0.2],[0.4, 0.4, 0.8],[0.1, 0.8, 0.3],[0.9, 0.9, 0.3]])
```

The array *im* represents an image with RGB (red, green, blue) values, where numbers range from 0 to 1. Write a program that creates a new numpy array representing an image in grayscale, where the pixel values are the minima of the respective red, green and blue values.

The program should print the new numpy array:

```
[[0.2 0.5 0.1 0.3]
 [0.2 0.4 0.1 0.3]]
```

Your program should still work if the original image array had different dimensions and pixel values. You are not allowed to use any pre-existing function that returns a minimum (eg. `numpy.min()`, `min()`).

## Question 6

```
hand_wash = 'You wash your hands properly by first wetting your
hands under running water, soaping and rubbing your hands together
until you get a lather. Rinse your hands thoroughly with running
water. Dry the hands, with a clean towel, if possible a disposable
paper towel or a cloth roller towel.'
```

Write a program to find out, how many different words there are in the above string *hand\_wash*, which of the words occurs the most often and at which positions in the text the most common word occurs. A word is defined here as a group of letters of the English alphabet without any special characters, separated from the next word by a space. Positions start with 0, so the 0<sup>th</sup> word is *You*. Please note that *You* and *you* should be counted as the same word. You may assume that there is only one most common word.

Next, generate a copy of the string *hand\_wash*, where the most common word as well as the word preceding the most common word at each of the occurrences is highlighted by capital letters. Everything else including the punctuation marks should remain as in the original string.

Your program should thus generate the following output:

```
Number of different words: 33
Most common word: hands
Its positions in the text: [3, 9, 17, 26, 33]
You wash YOUR HANDS properly by first wetting YOUR HANDS under
running water, soaping and rubbing YOUR HANDS together until you get
a lather. Rinse YOUR HANDS thoroughly with running water. Dry THE
HANDS, with a clean towel, if possible a disposable paper towel or a
cloth roller towel.
```

The program should be general enough that it would still work if the string *hand\_wash* was replaced by another string of English text. You may assume that the text does not contain any additional difficulties when compared to the text above (eg. the text will not start with the most common word).

## Question 7

Assume you are interested in better understanding the microbial composition in different human body sites. The file *microbial\_samples.txt* contains a dataset with abundance counts for 14 microbes from 30 microbial samples for 3 human body sites: the oral tract (14 samples), the gut (7 samples) and the skin (9 samples). Each line in the dataset contains the results (space separated) on one sample. Below, a subset of the dataset (one sample from each body site) is shown (list of strings *subset* in the file *for\_copying.py*).

```
Oral 1 0 351 0 0 258 1608 0 0 0 0 0 0 0
Gut 0 0 0 2 0 0 0 16 0 3974 0 0 153 676
Skin 0 280 0 0 2 1 0 0 0 0 0 0 0 0
```

The first word indicates the body site from which the sample was taken. The following numbers are abundance counts for the 14 microbes measured within each sample. The microbe identifiers for the 14 microbes are indicated in the following list of strings:

```
microbe_ids = ['S97-ac103', 'S97-ga105', 'S97-cy13', 'S97-ac137', \
               'S97-de22', 'S97-de227', 'S97-cy30', 'S97-fi362', 'S97-ga404', \
               'S97-fi47', 'S97-ac51', 'S97-ac769', 'S97-fi77', 'S97-ga86']
```

For example, in the above subset the second microbe (named 'S97-ga105') was found in the Skin sample (abundance count 280) but was absent in the other two samples (abundance counts 0).

Write a program to find the sample that harbors the most microbes in total (sum of all abundances). Print this maximum abundance sum as well as the body site the sample originates from.

Next, find for each body site the three microbes that are on average the most abundant (mean across all samples from one body site). Print these numbers in combination with the corresponding microbial identifiers in descending order for all 3 body sites.

Last, compare the most abundant microbes across the 3 body sites. Are there microbes that are among the top 3 most abundant microbes in more than one body site? If yes, print the microbe(s) and the body sites.

To indicate the expected output format, the output based on the above subset is shown:

```
The sample with the most microbes, 4821 counts, comes from the Gut
The top 3 on average most abundant microbes per body site:
Oral: S97-cy30 1608.0 S97-cy13 351.0 S97-de227 258.0
Gut: S97-fi47 3974.0 S97-ga86 676.0 S97-fi77 153.0
Skin: S97-ga105 280.0 S97-de22 2.0 S97-de227 1.0
S97-de227 is among the most abundant microbes in Oral and Skin
```

Your output should look the same but contain the results from analyzing the original dataset found in the file *microbial\_samples.txt*.

Your program should still work if the input files contained information about a different number of samples and microbes.

Please note, that the points you get for this question are largely based on your printed output and you will be given 0 points if the program prints nothing. In case not all the printed values are correct, you may still get part of the points.