Final exams: Programming 1 Introduction to Python

Instructions:

2 students write a program that should contain at least:

- Writing (defining) a function.
- Call for that function from the actual program.
- Proper syntax and use of (while, for) loop, if, elif, print, import math etc.
- Proper order of the code and lay-out (proper indents etc.)
- Ask for necessary interactive input via the Shell.
- Flexibility.
- Test the program and adjust when necessary (in case errors occur or inconsistent behaviour). Advice: test it multiple times and try to make it fail proof.
- One can use info there is on internet & self-written programs, etc.
- Spend in total 4 hours for each student.

Hand in:

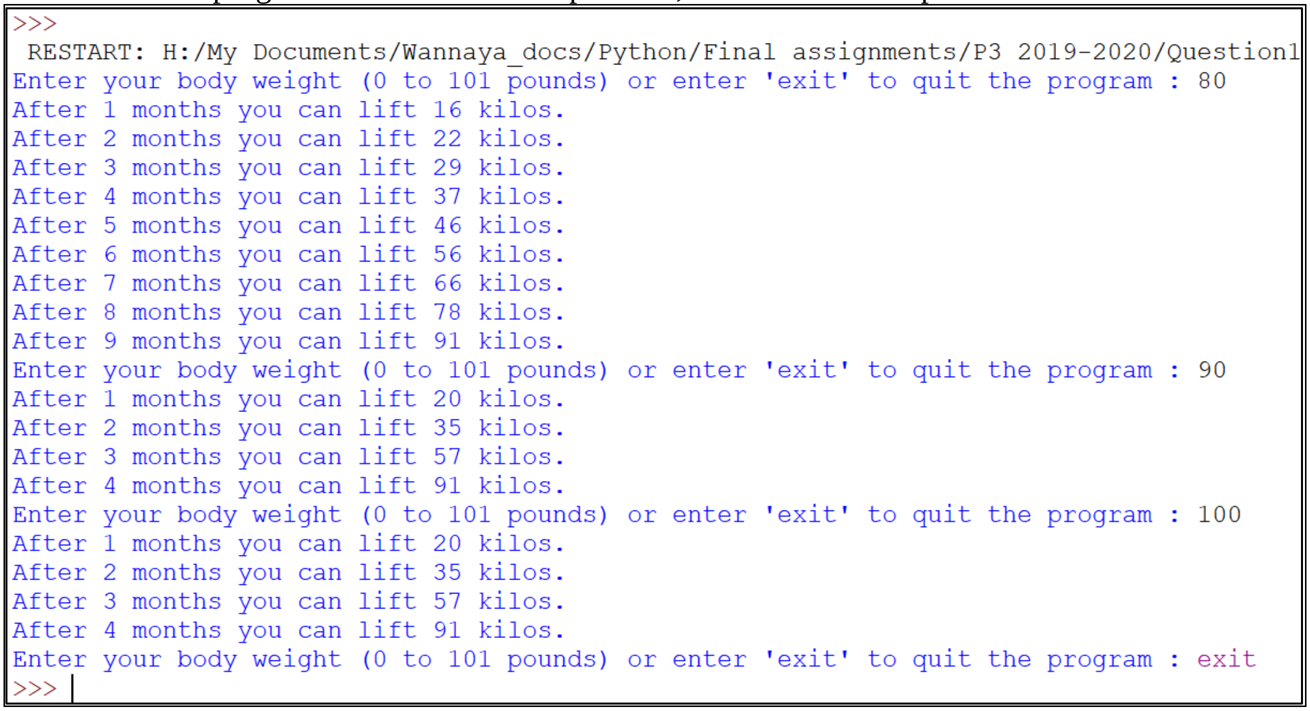
- Hand in via mail to your lecturer who taught you this term 3; that's: (613137@student.inholland.nl (Camilla Pistor, assistant) and also cc an email to Wannaya.besngamkham@inholland.nl) in 3 steps:
    1. Copy the final Python program-file into a word (*.docs)-file and attach it to the mail.
       Why do these laborious steps need to be taken? It's a work-around, since Python (*.py)-files cannot be transferred via mail since it is taken as 'malware' (a potential threat).
    2. Make a printscreen of the Shell via key: [CTRL-PrintScreen] of interactive input&output and copy it via key: [CTRL-v] this in the mail itself.
    3. Put both student number in the topic of your mail and state: 'Programming 1 assignment".

Question 1 A body weight forecasts **[45 points]**

1. **[5p]** Write a program where you ask the user for a body weight.
2. The program then forecasts how long it will take in months for the person to lift 100 kilograms. (**[5p]** for the proper application of the factor in items 5, 6, 7 and the initial value in 8 and 9).
   However, the person trains for up to a year.
3. The prognosis therefore stops at the target weight of 100 kilos or after a maximum of one year (12 months). (**[5p]** for an application of a for-while loop to create a correct number of iterations)
4. **[5p]** The forecast takes place in iterations of 1 month per iteration.
   The weight that a person can lift depends on their body weight.
5. **[5p]** If someone weighs between 0 and 51 pounds, the factor is 1.0
6. **[5p]** If someone weighs between 51 and 81 pounds, the factor is 1.1
7. **[5p]**If someone weighs between 81 and 101 pounds, the factor is 1.5
8. Each month, a person can lift 5 pounds more than the previous month.
9. All persons start with a minimum lifting weight of 10 kilos.
10. **[5p]**As long as the entry of body weight is not equal to **'exit'**, the program will continue to request new entries for the next forecast.

[5p] for a correct initial value.

Create a program to solve the above problem, as shown in example runs below.



```
>>> 
 RESTART: H:/My Documents/Wannaya_docs/Python/Final assignments/P3 2019-2020/Question1
Enter your body weight (0 to 101 pounds) or enter 'exit' to quit the program : 80
After 1 months you can lift 16 kilos.
After 2 months you can lift 22 kilos.
After 3 months you can lift 29 kilos.
After 4 months you can lift 37 kilos.
After 5 months you can lift 46 kilos.
After 6 months you can lift 56 kilos.
After 7 months you can lift 66 kilos.
After 8 months you can lift 78 kilos.
After 9 months you can lift 91 kilos.
Enter your body weight (0 to 101 pounds) or enter 'exit' to quit the program : 90
After 1 months you can lift 20 kilos.
After 2 months you can lift 35 kilos.
After 3 months you can lift 57 kilos.
After 4 months you can lift 91 kilos.
Enter your body weight (0 to 101 pounds) or enter 'exit' to quit the program : 100
After 1 months you can lift 20 kilos.
After 2 months you can lift 35 kilos.
After 3 months you can lift 57 kilos.
After 4 months you can lift 91 kilos.
Enter your body weight (0 to 101 pounds) or enter 'exit' to quit the program : exit
>>> |
```

Question 2 Compute the information content and entropy of data **[50 points]**

In information theory, you are often asked to compute the information content and the entropy of data. (Entropy is the average amount of information provided per event.)

Write a program that:

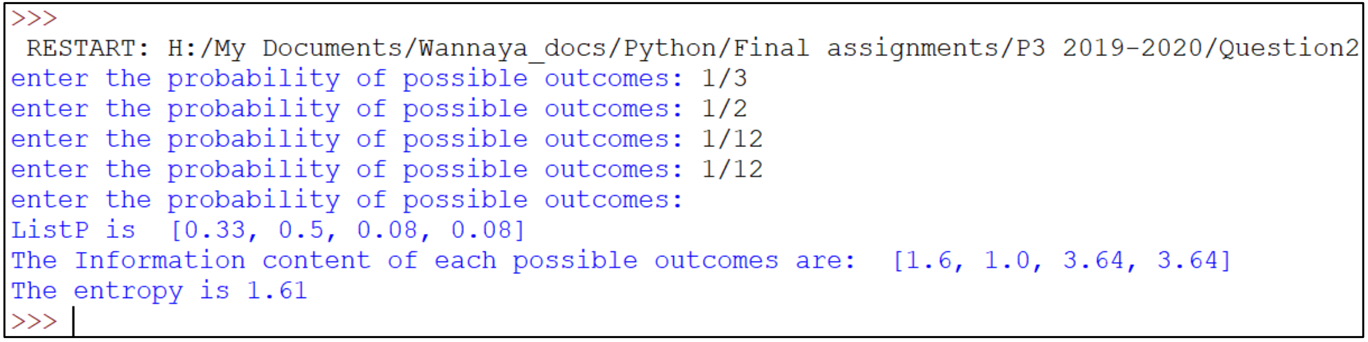
1. **[5p]** prompts the user to enter the probability of possible outcomes, *p*(*x*$_j$) and store those *p*(*x*$_j$) using a list,
2. **[5p]** displays the list, after user enter an empty data.
3. **[5p]** displays the information content:
   Formular:
   $$I(x_j) = \log_2\left(\frac{1}{p(x_j)}\right) = -\log_2 p(x_j)$$
4. **[5p]** displays the entropy of the data:
   Formular:
   $$H(x_j) = \sum_{j=1}^{n} p(x_j)\log_2\left(\frac{1}{p(x_j)}\right)$$
   where n is the total number of possible outcomes.
5. Your program should contain the following function:

   [10p]# Compute the information content from a list of data probabilities.
   ```
   def InfoContent(x):
   ```

   [10p]# Compute the entropy of data.
   ```
   def entropy(x):
   ```
6. **[10p]** your program show using of both function in item 5.

Note: the calculation result needs only floating number with 2 digits.

Create a program to solve the above problem, as shown in example runs below.



```
>>>
 RESTART: H:/My Documents/Wannaya_docs/Python/Final assignments/P3 2019-2020/Question2
enter the probability of possible outcomes: 1/3
enter the probability of possible outcomes: 1/2
enter the probability of possible outcomes: 1/12
enter the probability of possible outcomes: 1/12
enter the probability of possible outcomes:
ListP is  [0.33, 0.5, 0.08, 0.08]
The Information content of each possible outcomes are:  [1.6, 1.0, 3.64, 3.64]
The entropy is 1.61
>>> |
```