Lab_5

July 29, 2019

0.1 Run-length encoding

Write a program that implements a simple form of run-length encoding. Accepted input characters are all characters from 'a' to 'z'. The expected output is `<number><character>`, repeated for each run of characters. Here `<number>` is the amount of times `<character>` is repeated in sequence. Some examples:

```
aaeeeeae = 2a4e1a1e rr44errre = invalid input eeeeeeeeeeeeeeeeeeee = 21e
```

The program should take 'uncompressed data' as input. It should then compress the data and store the compressed data in two corresponding strings or lists. Finally, it should print the compressed data. Examples of the program are shown below:

```
Enter the data to be compressed: aaabbssssaad
The compressed data is: 3a2b4s2a1d
Enter the data to be compressed: aaAbbcccddd
error: invalid input
```

1 Weave 1

The goal of this exercise is to weave two series of numbers into a single series. The input consists of two lines of exactly ten numbers, which need to be woven into one. To weave the two series into one, consecutively take one number from the first series, followed by a number from the second series. Do not use a classyet. The example below illustrates the desired behaviour.

```
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
1 10 2 9 3 8 4 7 5 6 6 5 7 4 8 3 9 2 10 1
```

1.1 Weave 2

In this exercise, the weaving process is programmed into a NumberRow class. For this exercise create a class NumberRow which at least includes a method weave(number_row), which weaves one NumberRow with another. The problem to solve is extended to include a third series of numbers. The goal of this exercise is to weave the first and second series into a single series and consecutively weave a third series through the resulting series. The example below illustrates the desired behaviour.

```
1 2 3 4 5 6 7 8 9 10
10 9 8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
1 1 10 2 2 3 9 4 3 5 8 6 4 7 7 8 5 9 6 10 6 11 5 12 7 13 4 14 8 15 3 16 9 17 2 18 10 19 1 20
```

2 BodyMassIndex

Professor Hatzelklatzer has researched the extremely rare Hatzelklatzer-sydrome. There appear to be less cases of the sydrome in odd months than in even months. Further research should reveal if the syndrome affects people more often if they are too heavy. A way of determining whether someone is too heavy is the body-mass index (BMI). This is a measure of a person's weight taking into account their height. The BMI is defined as weight=length2. The World Health Organization (WHO) considers a BMI between 18,5 and 25 as ideal and considers people with such a BMI healthy. The program receives input consisting of two persons with their name, sex, length and weight.

```
Dean Johnson M 1.78 83
Sophia Miller V 1.69 60
```

Process this input into structured data. To achieve this, use an useful class with useful methods to enhance the structure of the program. Use this structured data to print for each person: an appropriate style of address, surname, the BMI and a statement whether this is considered healthy or not. Example

```
Mr. Johnsons BMI is 26.2 and is unhealthy.
Mrs. Millers BMI is 21.0 and is healthy.
```

3 BodyMassIndex2

Professor Hatzelklatzer has realized that although the program written in the BodyMassIndex assignment provides some usefull information on the effects of the BMI on the Hatzelklatzer syndrome, its output is quite difficult to process. To make the output easier to understand, the program should be adapted to analyze a group of test subjects. The program should read the input, determine the average BMI and count the number of cases of the Hatzelklatzer syndrome amongst people with a lower than average BMI and amongst people with a higher than average BMI. To this end, Professor Hatzelklatzer has provided you with the information of his diagnoses of all the test subjects. The input of the program now looks like this:

```
Dean Johnson M 1.78 83 Yes
Sophia Miller V 1.69 60 No
...
```

Just as in the previous BMI excercise, the input consists of people with their name, sex, length and weight. In addition, the word "Yes" has been added if the person suffers from the Hatzelklatzer syndrome, whilst "No" is added if the person did not suffer from the Hatzelklatzer syndrome. Instead of only two people, the input now consists of an unknown number of people, who all need to be analyzed. The input file can be found on Canvas. The output should look like this:

```
The average BMI of the test subjects is x.
There are y cases of the syndrome amongst people with a BMI >= x.
There are z cases of the syndrome amongst people with a BMI < x.
```