Beginselen van Programmeren
Exercise Session 2: Decisions, selections and simple data types

This exercise session will handle some simple data types, selection and decision making from chapter 3 of the handbook.

1 Warm up exercise

Warm up exercise: Answer this question on paper and verify with the computer. What is the value of each variable after the `if` statement?

1. 1)

```
n = 1
k = 2
r = n
if k < n :
        r = k
```

1. 2)

```
n = 1
k = 2
r = 0
if n < k and not r :
        r = k
else :
        r = k + n
```

1. 3)

```
n = 1
k = 2
r = k
if r < k:
        n = r
elif n < k :
    n = k + n
else :
        k = n
```

1. 4)

```
n = 1
k = 2
r = 3
if r < n + k or not r + n <= 2 * k:
        r = 2 * n
else :
        k = 2 * r
```

1. 5)

```
true = False
false = True
p = True
if not true and not not false:
    true = not true
    p = false and true
else:
    false = not false
    p = False or true
```

2 What is the difference? (R3.2)

Explain the difference between

```
s = 0
if x > 0 :
        s = s + 1
if y > 0 :
        s = s + 1
```

and

```
s = 0
if x > 0 :
        s = s + 1
elif y > 0 :
        s = s + 1
```

3 Float describer (P3.2)

Write a program that reads a floating-point number and prints `zero` if the number is zero. Otherwise, print `positive` or `negative`. Add `small` if the absolute value of the number is less than 1, or `large` if it exceeds 1.000.000. First, create a flowchart describing the algorithm, then translate this flowchart to pseudo code and finally to Python code.

```
Input a floating-point number: 0.232
This number is a small positive number.
Input a floating-point number: 1.232
This number is a positive number.
Input a floating-point number: -0.232
This number is a small negative number.
```

```
Input a floating-point number: -1.232
This number is a negative number.
Input a floating-point number: 1220000
This number is a large positive number.
Input a floating-point number: 0
This number is zero.
```

4 Bull's eye (P2.26)

Draw a *bull's eye* – a set of concentric rings in alternating black and white colors (see page 83). Also draw a bounding box around the bull's eye.

5 Sorting

Write a program that reads in three integer numbers and prints the three inputs in sorted order. The program always expects exactly 3 numbers as input. For example:

```
first number? 4
second number? 9
third number? 2

The inputs in sorted order are: 2, 4 and 9
```

6 Letters to grades (P3.12)

Write a program that translates- a letter grade into a number grade. Letter grades are A, B, C, D and F, possibly followed by + or −. Their numeric values are 4, 3, 2, 1, and 0. There is no $F+$ or $F-$. A plus (+) increases the numeric value by 0.3, a minus (−) decreases it by 0.3. However, an $A+$ always has the value 4.0.
Example:

```
Enter a letter grade:
B-
Numeric value: 2.7
```

Extra: Extend the program so that it may also perform conversions the other way around, that is, given a number grade, the program converts the grade into a letter grade.

7 Float comparator (P3.21)

Write a program that reads in two floating-point numbers and tests whether they are the same up to two decimal places. Here are two sample runs.

```
Enter a floating-point number: 2.0
Enter a floating-point number: 1.99998
They are the same up to two decimal places.
Enter a floating-point number: 2.0
Enter a floating-point number: 1.98999
They are different.
```

8 Quadratic equation

Write a program that prints all real solutions to the quadratic equation $ax2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions. Examples:

```
Enter coefficient a: 1
Enter coefficient b: 4
Enter coefficient c: -5
Has solution: YES
Solution 1: 1.0
Solution 2: -5.0
Enter coefficient a: 2
Enter coefficient b: 2
Enter coefficient c: 4
Has solution: NO
```

9 Leap year (P3.27)

A year with 366 days is called a leap year. Leap years are necessary to keep the calendar synchronized with the sun because the earth revolves around the sun once every 365.25 days. Actually, that figure is not entirely precise, and for all dates after 1582 the Gregorian correction applies. Usually years that are divisible by 4 are leap years, for example 1996. However, years that are divisible by 100 (eg. 1900) are not leap years, but years that are divisible by 400 are leap years (eg. 2000). Write a program that asks the user for a year and computes whether that year is a leap year in the Gregorian calendar. Use a single `if` statement and Boolean operators. Note that the Gregorian calendar was introduced in 1582. Example:

```
Enter the year you wish to check: 1994
1994 is not a leap year.
Enter the year you wish to check: 1745
1745 is not a leap year
Enter the year you wish to check: 1840
1840 is a leap year
Enter the year you wish to check: 1600
1600 is a leap year
Enter the year you wish to check: 1745
1745 is not a leap year
```

10 Income taxes (P3.23, P3.25)

10.1 Single schedule

The original U.S. income tax of 1913 was quite simple. The tax was:

- 1 percent on the first $50,000 of your annual personal income.
- 2 percent on the amount over $50,000 up to $75,000 of your annual personal income.
- 3 percent on the amount over $75,000 up to $100,000.
- 4 percent on the amount over $100,000 up to $250,000.

- 5 percent on the amount over $250,000 up to $500,000.
- 6 percent on the amount over $500,000.

There was no separate schedule for single or married taxpayers. Write a program that asks the user his annual personal income and then computes the income tax based on this schedule. Example:

```
What was your annual income? 245785
On 245785.00 income you have to pay 7581.40 taxes
```

10.2 Marital status

Write a program that computes the taxes for the schedule on page 146 of the book (exercise P3.25). **Hint:** First create a flowchart on paper visualizing the control flow of the program.

11 Extra exercises

P3.9, P3.15, P3.17, P3.28, P3.45