University of Huddersfield
School of Computing and Engineering

CFM2103 Mathematical Programming

Practical **Week 2**

Work through the questions included below. If you get stuck, please revisit the information on the slides used today before you ask for help.

---

1. The quadratic equation $ax^2 + bx + c = 0$ ($a \neq 0$) has three different types of solutions, depending on whether $\Delta > 0$, $\Delta = 0$ or $\Delta < 0$, respectively; here, $\Delta := b^2 - 4ac$ is the discriminant of the quadratic. More specifically,

$$x_{1,2} = \frac{1}{2a}(-b \pm \sqrt{\Delta})\,, \qquad \text{if} \quad \Delta > 0\,,$$

$$x_1 = x_2 = -\frac{b}{2a}\,, \qquad \text{if} \quad \Delta = 0\,,$$

$$x_{1,2} = \frac{1}{2a}(-b \pm \mathrm{i}\sqrt{-\Delta})\,, \qquad \text{if} \quad \Delta < 0\,.$$

   (a) Download the code `Week2Ex1.py` and study it. This computer program is for solving the linear equation $ax + b = 0$ ($a \neq 0$). Note that it contains a conditional statement because one needs to ensure that 'a' is not zero.

   (b) Based on the first part and the formulae outlined above, write a piece of code that solves a quadratic equation. The coefficients a, b, c should be taken interactively from the user (as in some of the examples seen last week), and the corresponding roots must be displayed neatly on the screen. As you need to check the sign of the discriminant, your code will have to contain a conditional statement (with three "branches"); this will have to be nested inside an outer conditional statement that checks if the leading coefficient in the quadratic is zero or not.

2. Use for-loops to generate the three patterns indicated below (one piece of code for each pattern). The dimensions of these patterns are arbitrary and must be (interactively) specified by the user; you are allowed to print only one asterisk at a time.

```
****
****
****
```

```
*
**
***
****
```

```
_*_*_
_*_*_
_*_*_
```

3. The Syracuse sequence is generated by starting with a natural number and repeatedly applying the following function *until* reaching 1:

$$syr(x) = \begin{cases} x/2\,, & \text{if } x \text{ is even}\,, \\ 3x + 1\,, & \text{if } x \text{ is odd}\,. \end{cases}$$

For example, the Syracuse sequence starting with 5 is: 5, 16, 8, 4, 2, 1. Write a computer program that gets a starting value from the user and then prints the Syracuse sequence for that starting value.

4. *Heron's method* for finding an approximation for $\sqrt{S}$, where $S > 0$ is any real positive number is based on the following algorithm:

   (a) set x_0 to be a number that is a rough approximation of $\sqrt{S}$;

   (b) define x_1 to be the average of x_0 and S/x_0;

   (c) define x_2 to be the average of x_1 and S/x_1; in general x_{n+1} is defined to be the average of x_n and S/x_n;

   (d) the process is repeated until x_n is close enough to the actual value of $\sqrt{S}$.

   Use Python's `sqrt` function to evaluate the "exact" value of $\sqrt{2023}$, x_* (say), and then write a computer program that implements the above algorithm. The stopping criterion for the iterations in your code should be

   $$|x_n - x_*| < TOL\,,$$

   where TOL is a small number (e.g., 10^{-5}, 10^{-7}, etc) that should be chosen by the user. Your program should display on the screen the approximation found, the "exact" value mentioned above, as well as the number of iterations necessary to get the result.

Optional questions:

If you would like to receive additional feedback on your work, you should attempt the questions included below and submit your computer code in a zipped folder on Brightspace by no later than 5:00 PM next Tuesday.

1. The greatest common divisor (GCD) of two values can be computed using Euclid's algorithm. Starting with the values m and n, we repeatedly apply the formula

   $$n, m = m, n\%m$$

   until m is 0. At that point n is the GCD of the original m and n. Write a program that finds the GCD of two numbers using this algorithm. Test your code on several pairs of integers.

2. The area of a circle of radius $R > 0$ can be approximated by the area of a regular polygon of n sides inscribed in that circle. The area A of such a polygon is given by the formula

   $$A = \frac{1}{2} n R^2 \sin\left(\frac{2\pi}{n}\right)\,.$$

   Write a computer code that, for a given $R > 0$, determines the minimum number of sides of the above polygon such that the approximation is correct to 6 significant digits. Test your code for different values of R.

   [Hint: If $A_c \equiv \pi R^2$, your stopping criterion should be $|A - A_c| < 10^{-6}$.]