Workshop

Week 38

Exercise 1a

Modify the random number generator (exercise 1) from last workshop so that the program now *draws multiple random numbers* within the given bounds.

The program should now:

- prompt the user for a lower and upper bound, and for the number of draws.
- use a **while loop** to draw and display the random numbers to the user.

For simplicity, you can ignore checking whether the inputs are valid.

1. Write an algorithm for the solution.

2. Implement the solution using Python.

Exercise 1a

Solution proposal for the algorithm:

Input: upper, lower, no_draws
Output: random numbers

```
i ← 0
while i < no_draws do
    Draw random number between upper and lower
    Print random number
    i ← i + 1
end while
```

Exercise 1b

Modify the random number generator (exercise 1) from last workshop so that the program now *draws multiple random numbers* within the given bounds.

The program should now:

- prompt the user for a lower and upper bound, and for the number of draws.
- use a **for loop** to draw and display the random numbers to the user.

For simplicity, you can ignore checking whether the inputs are valid.

1. Write an algorithm for the solution.

2. Implement the solution using Python.

Exercise 1b

Solution proposal for the algorithm:

Input: upper, lower, no_draws
Output: random numbers

```
for i ← 1 to no_draws do
    Draw random number between upper and lower
    Print random number
end for
```

Exercise 2

Use the solution proposal for the prisoner’s dilemma (exercise 3) from last workshop and modify the program so that it does not terminate if the players enter invalid inputs (i.e. not «1» or «2»).

Modify the program so that the players are re-prompted for inputs until they enter valid inputs.

1. Ask the players for their choices, and if their choices are not valid, ask for new choices in a **while loop**.
2. Modify the program so that the **while loop** uses a **boolean flag**.

Exercise 3

Again modify the random number generator so that the program *continues to draw random numbers until the user decides to quit.*

The program should now:

- initialize a **while loop** that in each iteration:
  - prompt the user for a lower and upper bound.
  - draw and display the random number.
  - ask the user whether to continue or quit.

For simplicity, you can ignore checking whether the inputs are valid.

Algorithm:

```
while The user does not quit do
    Get lower and upper bound from user
    Draw random number between bounds
    Print the random number to user
    Ask if user wants to quit or not
end while
```