Note: For every exercise, create ~~an own project~~. Exercises which are not finished in class must be completed at home!

Learning outcome of this set of exercises: Using the development environment, simple I/O operations, control elements, elementary data types.

Exercise categories:

A – very basic, intended for inexperienced developers

B – fair, a little bit more complex but still for starters

C – challenging, complexity is higher, additional programming constructs may be required

Exercise 1 – Metric and English units (A)

Develop a program which translates meter units into miles and feet.

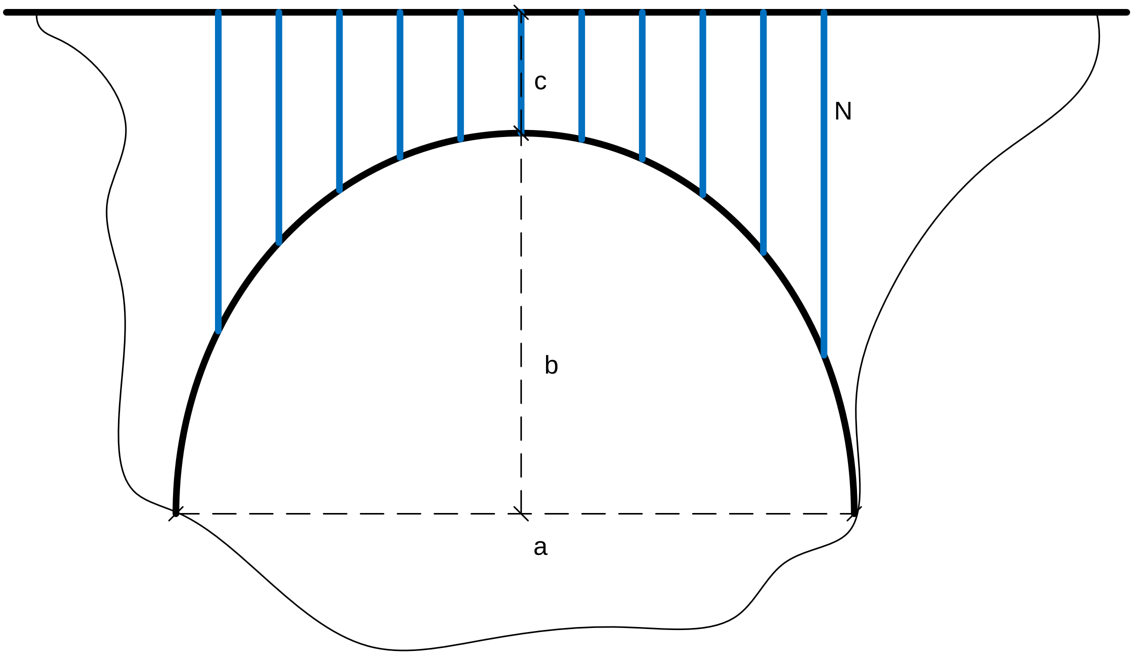
- The user is prompted to enter a value in meter
- In case the value is negative, an error message is printed and the input is repeated (hint: use a ~~do~~ while loop and if statement for this)
- The value is printed in miles and feet. 1 mile = 1852m and 1 foot = 0.3048m

Example output:

```
Metric to English Converter
Please enter a value in meter: -4
only positive values are allowed!
Please enter a value in meter: 1000

1000m = 0.539957 mile = 3280.84 ft
```

Exercise 2 – Material for a bridge (B)

Your task is to calculate the material required for the N vertical elements of a steel bridge.



The bow of the bridge has the form of a parabola with a width of ‘~~b~~’ and a height of ‘~~a~~’. The street is located with the distance ‘c’ above the bow. ‘N’ vertical elements attach the street to the bow. The bow is represented by the formulas

$$y(x) = p1 * x^2 + p2 * x + p3$$

$$p1 = -\frac{4 * b}{a^2}$$

$$p2 = \frac{4 * b}{a}$$

$$p3 = 0$$

Write a program performing the following tasks:

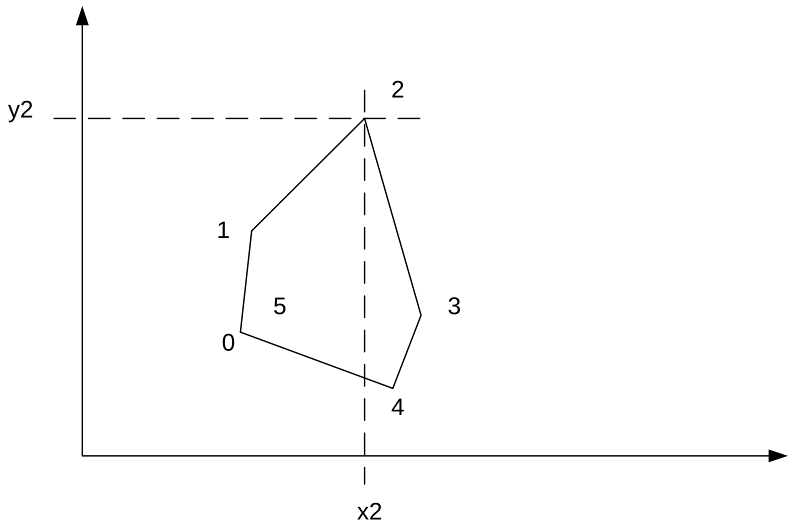
- ~~Ask the user to enter the values for a, b, c and N~~. Check that all values are bigger than 0. ~~Illegal values shall be entered again~~.
- Calculate and print the length of every steel element as well as the total length of all steel elements
- Tricky: The first steel element has to be at position 0, the last at position a!

Example output:

```
Calculation bridge material

~~Enter the value for a (width of the bow):~~ 10
~~Enter the value for b (heigth of the bow):~~ 2
~~Enter the value for c (heigth of the street):~~ 1
~~Enter the value for N (numer of vertical carrier elements):~~ 7

Calculation:
  Element No. 0 at x-position 0 has a length of 3
  Element No. 1 at x-position 1.66667 has a length of 1.88889
  Element No. 2 at x-position 3.33333 has a length of 1.22222
  Element No. 3 at x-position 5 has a length of 1
  Element No. 4 at x-position 6.66667 has a length of 1.22222
  Element No. 5 at x-position 8.33333 has a length of 1.88889
  Element No. 6 at x-position 10 has a length of 3


Total length of material: 13.2222
```

Exercise 3 – Area of a Polygon (B)

Calculating the area of a triangle and a square is pretty simple, but how about a polygon shape with 5 or more corners? Of course you can split every polygon shape into a set of triangles, but this is pretty error prone of you do this manually. The Gaussian formula for integrals is based on the triangle approach but can easily be automated.



$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i \cdot y_{i+1} - x_{i+1} \cdot y_i)$$

Write a program to calculate the area of a polygon:

- ~~The maximum number of corners is 10~~
- The user enters the x and y positions of every corner one after the other
- If the user enters ~~the first coordinate again~~, the polygon shape is finished and the calculation starts
- If 10 coordinates have been entered and the user did not enter the first coordinate again, the program prints a warning, the last coordinate is set to the value of the first coordinate and the calculation starts
- If less than three coordinates are entered, an error message is printed and the program terminates

Some hints:

- Use ~~array~~s to store the x and y coordinates
- ~~Check the difference between a bitwise AND operator (&) and a logical AND operator (&&)~~
- You can use the break command to exit a loop
- Sometimes, the calculated area is negative. Why? How can you correct this?

Output:

```
Calculation of a polygon area using Gauss

Enter the coordinates for corner 0 : 1 1
Enter the coordinates for corner 1 : 2 1
Enter the coordinates for corner 2 : 2 2
Enter the coordinates for corner 3 : 1 2
Enter the coordinates for corner 4 : 1 1


Calculated area: 1
```