Valentin Helml
Dirk Praetorius
Julian Streitberger

Summer term 2022
29 April 2022

Exercise
Scientific programming in mathematics

Series 7

Starting with this exercise sheet, all exercises should be programmed in the C++ programming language. For more information about the class string, please refer to the webpage String.

Exercise 7.1. Any integer $0 \leq n \leq 2^{16} - 1 = 65535$ has a binary representation of the form

$$n = \sum_{n=0}^{15} a_i 2^i$$

with coefficients $a_i \in \{0, 1\}$ for $i = 0, \dots, 15$. Implement the following two functions:

- `string dec2bin(int n)`, which, given an integer $0 \leq n \leq 65535$, computes and returns a string with the representation in the binary numeral system,
- `int bin2dec(string s)`, which, given a string consisting of at most 16 characters containing a binary representation of a number between `0` and `1111111111111111`, returns the representation of the integer in the decimal system.

The functions work with binary representations without leading zeros. For instance, for $n = 77$, the function `dec2bin` should return the string `1001101`. Test your implementation appropriately! Save your source code as `decVSbin.cpp`.

Exercise 7.2. A palindrome is a word, which reads the same backward and forward, e.g., radar, level, madam, racecar. Write a function `bool isPalindrome(string word)` that checks whether a word is a palindrome. The function shall return the value `true` if the input string is a palindrome, otherwise it shall return the value `false`. Write a main program, which reads a word from the keyboard and checks whether it is a palindrome. Save your source code as `palindrome.cpp`.

Exercise 7.3. The numeral system of Roman numerals is an additive system such that any symbol has a fixed value: $I = 1$, $V = 5$, $X = 10$, $L = 50$, $C = 100$, $M = 1000$. In order to avoid four characters being repeated in succession, the following subtractive notation is used: $IV = 4$, $IX = 9$, $XC = 90$, $CD = 400$, $CM = 900$. Apart from this, the value of a symbol is independent of its position. Implement the following two functions:

- `string int2roman(int n)`, which computes and returns the reprensation of an integer $1 \leq n \leq 3999$ as a Roman numeral,
- `int roman2int(string s)`, which computes and returns the decimal representation of a Roman numeral between I and $MMMCMXCIX$.

Test your implementation appropriately! Save your source code as `roman.cpp`.

Exercise 7.4. Write a class `University` to store some information about a university. The class should contain the variable members `name` (`string`), `city` (`string`), and `num_students` (a non-negative `int`). All data members should be declared as private. Therefore, in order to work with the class, you have to implement suitable access methods (`get` and `set` methods). Moreover, implement the methods `void graduate()`, and `void newStudent()`. If the method `graduate` is called, the number of students gets decreased by one, whereas if `newStudent` is called, the number of students increases by one. Use `assert` to ensure that `graduate` does not make `num_students` negative. Test your implementation with suitable examples! Save your source code as `university.{hpp,cpp}`.

Exercise 7.5. Write a class `Name` which contains two members, `firstname` and `surname`, both of type `string`. Implement the standard access methods to work with the class. Moreover, implement the set method `setFullName`, which has one string variable as input parameter and splits the input in first name and surname automatically. Note that the input can contain multiple first names (but you can assume that the surname consists of only one word). Furthermore, write a method `printName()`, which prints the name to the screen. In case of multiple first names, only the first one should printed entirely. For all others, only the initial should be printed. For example, for the name John Johnny Doe, the method `printName` should print the output John J. Doe to the screen. Save your source code as `name.{hpp,cpp}`.

Exercise 7.6. Write a class `Customer` for a bank customer. The class contains the name of the customer as `string`, the current balance in EUR as `double` and a PIN code as `int`. Implement the access functions (`set` and `get` methods) for the member variables together with the following methods:

- `void printBalance()`, which prints the current balance to the screen.
- `bool CheckPIN()`, which reads a PIN code from the keyboard and checks whether it is correct or not.
- `void drawMoney(double amount = 0)`, which reads a PIN code from the keyboard. If the PIN is correct, the method needs to know the amount of money the customer wants to withdraw. This is either passed to the method as optional input parameter or read from the keyboard. If the desired amount is positive and not larger than the current balance, the operation is performed and the new balance is printed to the screen. If the new balance becomes less than 10.00 EUR, the methods prints a warning to the screen. If the desired amount is larger than the current balance, then the operation is interrupted.

Use `assert` to intentionally terminate the program and return error messages whenever needed (e.g., when the PIN is wrong or when the desired amount is not admissible). Save your source code as `customer.{hpp,cpp}`.

Exercise 7.7. An ellipse is a curve in the plane surrounding two focal points such that the sum of the distances to the two focal points is constant for every point on the curve. If Cartesian coordinates are introduced such that the axes of the ellipse are parallel to the coordinate axes, the points of an ellipse with center $(x_0, y_0) \in \mathbb{R}^2$ and semi-axis lenghts $a, b > 0$ satisfy the equation

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1.$$

Recall that an ellipse is the generalization of a circle, in the sense that a circle of radius r is an ellipse with semi-axis lengths $a = b = r$. Write a class `Ellipse` to store the ellipses of the aforementioned type. The class contains the variable members `center`, a two-dimensional array of type `double` containing the coordinates of the center, as well as the variables `a` and `b`, both of type `double`, containing the lengths of the semi-axes. Implement the standard access methods (`get` and `set` methods) to work with the class. Moreover, implement the following methods:

- `bool isInside(double x, double y)`, which checks whether a given point lies in the part of the plane delimited by the ellipse. If the point lies exactly on the ellipse, the return value of the method is still `true` and a message informing the user about this fact is additionally printed to the screen.

- `bool isCircle()`, which checks whether an ellipse is actually a circle.

- `printFocalPoints()`, which computes and prints the position of the focal points to the screen. Note that, if $a > b$, the focal points are located on the axis of the ellipse that is parallel to the x-axis and their coordinates are given by $(x_0 \pm c, y_0)$, where $c := \sqrt{a^2 - b^2}$. Adapt these formulae accordingly for the remaining cases $a \leq b$. What is the position of the focal points of a circle?

- `double getEccentricity()`, which computes and returns the eccentricity e of the ellipse given by

  $$e = \sqrt{1 - \left(\frac{\min\{a,b\}}{\max\{a,b\}}\right)^2}.$$

  What is the eccentricity of a circle?

Test your implementation in a suitable way! Save your source as `ellipse.{hpp, cpp}`.

Exercise 7.8. Write a class `matrix` to store sparse matrices. The class contains the number of rows `m`, the number of columns `n` and the number of non-zero entries `nnz` of type integer as well as `I` and `J` of type `int*`, an indicator whether the format is in coordinate or CCS format `isCoordinate` of type `bool`, and `A` of type `double*`; cf. Exercise 3.3 of sheet 3. Moreover, implement standard access methods (`get` and `set` methods) as well as the methods `sort`, which sorts a matrix given in coordinate format columnwise, `cco2ccs` and `ccs2cco`, which transfer the compressed coordinate format to the CCS format and vice versa. Test your implementation in a suitable way! Save your source as `matrix.{hpp, cpp}`.