Practice Exercises

Exercises

1. Write a program that takes as input two numbers and returns the greatest

2. Write a program that takes an integer number and returns the string “EVEN” if the number is even or “ODD” if the number is odd

3. Write a program that returns the team of a student given its identification number. There are three possible teams: RED, GREEN and BLUE. The assignment takes place with the following criterion: the student with id 1 goes to the RED team, id 2 to the GREEN, id number 3 to the BLUE, id 4 to the RED, id 5 to the GREEN etc. (Note: you can use case construction)

4. Write a program that takes as input an integer corresponding to the score of a student and returns "Insufficient" if it is less than 18, "Just enough" (if the score is 18), "Low" (if the score is between 19-20), "Medium" (if the score is between 21-23), "Good" (if the score is between 24-26), "High" (if the score is between 27-29), "Maximum" (if the score is 30) "Impossible" (in other cases) (Hint: you can use the guard construct)

5. Write a program that takes three coefficients a, b and c of a second degree equation, and returns the solutions if these are real; if they are not, it must simply return “Non-real values”.

6. A secular year (divisible by 100) is a leap year if it is divisible by 400, a non-secular year is a leap year if it is divisible by 4. For example, the year 1900 was not a leap year, 1996 was a leap year, 2000 was, 2002 was not. Write a program that takes a year as input and indicates whether it is a leap year or not.

7. Write a program that takes as input a list of integers and returns the average of the numbers in the list.

8. Write a program that takes a list of integers as input, and returns a list consisting first of all the even values in the order in which they are in the input list and then all the odd values in the reverse order.
   Example: given the values: 8 1 3 2 8 6 5, the
   program will return: 8 2 8 6 5 3 1

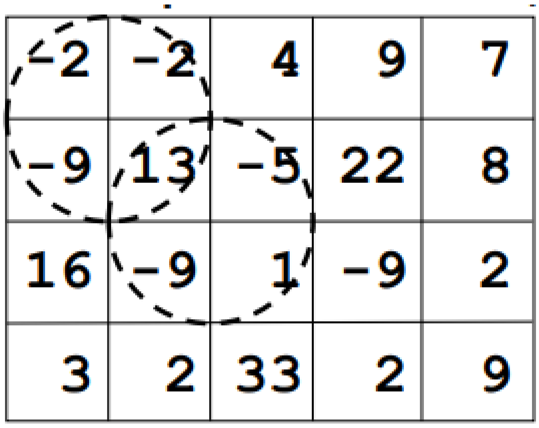
9. Write a program that takes a list of doubles as input, and returns a list of 3-value moving averages of these numbers. The program must check that the number of values in the list is at least equal to 3. The moving average is an arithmetic average over only a part of the values (in this case 3), for example if the sequence of values is given:
   2.1, 4.2, 1.3, 6.7, 3.1, 5.5, 2.1, 4.9, 3.0, 5.4, 3.9
   the program has to calculate the average of 2.1, 4.2 and 1.3 and record it, then the average of 4.2, 1.3 and 6.7 and record it, then 1.3, 6.7 and 3.1 and record it, etc. up to 3.0, 5.4 and 3.1

10. Write a program that takes a list of integer values as input and identifies the longest sequence of consecutive equal numbers. If several sequences of the same length are identified, consider only the first one identified. The program must indicate the repeated value and the number of occurrences of that value.
Example:
Inputs: 19, 3, 15, 15, 7, 9, 9, 9, 9, 12, 3, 3, 3
Output: number: 9, repetitions: 4

11. Write a program that takes as input a matrix of integers and returns the maximum, the minimum,the sum and the average.

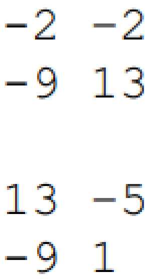
12. Write a program that takes a matrix of integer values as input and returns how many values are even and how many are odd.

13. Write a program that takes as input a matrix of integer values, and two integer values representing the sizes and returns all the submatrices of the given matrix whose elements sum is equal to zero and their size are equal to the given sizes (the first integer is the number of line and the second is the number of column).
Example:
Input



-2	-2	4	9	7
-9	13	-5	22	8
16	-9	1	-9	2
3	2	33	2	9

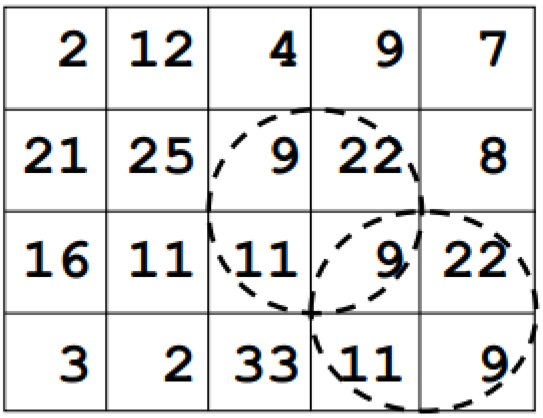
2 (number of rows) 2 (number of columns)
Output



```
-2 -2
-9 13


13 -5
-9 1
```

14. Write a program that takes two matrices of integer values as input and returns the list of coordinates where the second was found in the first.
Example:



2	12	4	9	7
21	25	9	22	8
16	11	11	9	22
3	2	33	11	9

output [(1, 2), (2, 3)]

15. Write a program that takes as input two matrices and returns the product of the two. The program must verify that the matrices are valide and if they can be multiplied. Two matrices can be multiplied if the number of columns of the first matrix is equal to the number of lines of the second.

16. Write a function that divides two integral numbers using recursive subtraction. The type should be (Integral a) => a -> a -> a. Redo this exercise using the type (Integral a) => a -> a -> (a, a) where (a, a) represents the quotient and the rest of the division.

17. Write a function that recursively sums all numbers from 1 to n, n being the argument. So that if n was 5, you'd add 1 + 2 + 3 + 4 + 5 to get 15.
The type should be (Eq a, Num a) => a -> a.

18. Write a function that multiplies two integral numbers using recursive summation. The type should be (Integral a) => a -> a -> a.

19. Write a program that takes two strings as input and returns the longest. The first if they are of equal length.

20. Write a program that takes two strings as input and returns the greater one.

21. Write a program that takes as input a string and counts the number of characters.

22. Write a program that takes a string as input, and returns the same string converted to all uppercase.

23. Write a program that takes a string as input and checks if it contains at least one 'A' among the first 10 characters.

24. Write a program that takes a string as input and counts how many digits it contains.

Example “Hello2022! C6? " must give 5.

25. Write a program that takes a string as its input and counts how many uppercase letters, lowercase letters, digits and other characters it contains.
    Example
    "Hello2022! C6? " must give:
    uppercase: 2, lowercase: 4, digits: 5, others: 4.

26. Write a program that takes two strings of different lengths as input and indicates whether the shortest is contained only once in the longest.

27. Write a function that tells you whether or not a given String (or generic list) is a palindrome. Here you’ll want to use a function called reverse, a predefined function that does what it sounds like.
    reverse :: [a] -> [a]
    reverse "blah"
    "halb"
    Example:
    radar, rotor, madam, kayak, anilina, otto, elle.

28. Write a program that checks if an input string is made up of two equal parts, ignoring the central character if the length is odd (eg “HelloHello”, “HelloXHello”).

29. Write a program that takes a sentence as input, and counts the number of words (sequences of letters of the alphabet) that compose it and the average length of the words themselves.
    Example
    If it is given as input:
    Yesterday ... I went to eat with friends!
    the program must indicate that there are 7 words and that the average length is 4.28 characters.

30. Write a program to calculate the square root using Newton's iterative formula:

    formula

    $$x_{i+1} = \frac{1}{2}\left(x_i + \frac{A}{x_i}\right)$$

    Given the value A, we want to calculate its square root x. The given formula calculates increasingly precise values of x. Initially we consider $x_{i=0}$ = A, obtaining a value x_1 which roughly approximates the value of the square root. We put x_1 back into the formula (in place of x_i) to get x_2 which is a better approximation than the previous one. This continues as long as the result varies (ie $x_{i-1} – x_i >$ DLB_EPSILON). DBL_EPSILON = 2.2204460492503131e-16

31. Write a program to compute e^x by means of its series development:

    $$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$$

Each fraction adds precision to the result, so it is better to use adequately high values of n, for example between 30 and 40. Check that the results calculated in this way are consistent with those provided by the intrinsic function exp by calculating the difference of the values.

32. Given a number, determine whether or not it is valid per the Luhn formula. The Luhn algorithm is a simple checksum formula used to validate a variety of identification numbers, such as credit card numbers and Canadian Social Insurance Numbers.

    The task is to check if a given string is valid.

    Validating a Number

    Strings of length 1 or less are not valid. Spaces are allowed in the input, but they should be stripped before checking. All other non-digit characters are disallowed.

    Example 1: valid credit card number

    4539 3195 0343 6467

    The first step of the Luhn algorithm is to double every second digit, starting from the right. We will be doubling

    4_3_ 3_9_ 0_4_ 6_6_

    If doubling the number results in a number greater than 9 then subtract 9 from the product. The results of our doubling:

    8569 6195 0383 3437

    Then sum all of the digits:

    8+5+6+9+6+1+9+5+0+3+8+3+3+4+3+7 = 80

    If the sum is evenly divisible by 10, then the number is valid. This number is valid!

    Example 2: invalid credit card number

    8273 1232 7352 0569

    Double the second digits, starting from the right

    7253 2262 5312 0539

    Sum the digits

    7+2+5+3+2+2+6+2+5+3+1+2+0+5+3+9 = 57

    57 is not evenly divisible by 10, so this number is not valid.

33. Write a robot simulator.

A robot factory's test facility needs a program to verify robot movements.The robots have three possible movements:

- turn right
- turn left
- advance

Robots are placed on a hypothetical infinite grid, facing a particular direction (north, east, south, or west) at a set of {x,y} coordinates, e.g., {3,8}, with coordinates increasing to the north and east.

The robot then receives a number of instructions, at which point the testing facility verifies the robot's new position, and in which direction it is pointing.

- The letter-string "RAALAL" means:
    - Turn right
    - Advance twice
    - Turn left
    - Advance once
    - Turn left yet again

- Say a robot starts at {7, 3} facing north. Then running this stream of instructions should leave it at {9, 4} facing west.

To complete this exercise, you need to create the data type Robot, and implement the following functions:

- bearing
- coordinates
- mkRobot
- simulate
- turnLeft
- turnRight

You will find a dummy data declaration and type signatures already in place, but it is up to you to define the functions and create a meaningful data type, newtype or type synonym.

34. Implement run-length encoding and decoding.

Run-length encoding (RLE) is a simple form of data compression, where runs (consecutive data elements) are replaced by just one data value and count.

For example we can represent the original 53 characters with only 13.

"WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWWWWWWWWWWWWB" -> "12WB12W3B24WB"

RLE allows the original data to be perfectly reconstructed from the compressed data, which makes it a lossless data compression.

"AABCCCDEEEE" -> "2AB3CD4E" -> "AABCCCDEEEE"

For simplicity, you can assume that the unencoded string will only contain the letters A through Z (either lower or upper case) and whitespace. This way data to be encoded will never contain any numbers and numbers inside data to be decoded always represent the count for the following character.

35. The ISBN-10 verification process is used to validate book identification numbers. These normally contain dashes and look like: 3-598-21508-8

ISBN

The ISBN-10 format is 9 digits (0 to 9) plus one check character (either a digit or an X only). In the case the check character is an X, this represents the value '10'. These may be communicated with or without hyphens, and can be checked for their validity by the following formula:

```
(x1 * 10 + x2 * 9 + x3 * 8 + x4 * 7 + x5 * 6 + x6 * 5 + x7 * 4 + x8 * 3 + x9 * 2 + x10 * 1) mod 11 == 0
```

If the result is 0, then it is a valid ISBN-10, otherwise it is invalid.

Example

Let's take the ISBN-10 3-598-21508-8. We plug it in to the formula, and get:

(3 * 10 + 5 * 9 + 9 * 8 + 8 * 7 + 2 * 6 + 1 * 5 + 5 * 4 + 0 * 3 + 8 * 2 + 8 * 1) mod 11 == 0

Since the result is 0, this proves that our ISBN is valid.

Task

Given a string the program should check if the provided string is a valid ISBN-10. Putting this into place requires some thinking about preprocessing/parsing of the string prior to calculating the check digit for the ISBN.

The program should be able to verify ISBN-10 both with and without separating dashes.

Caveats

Converting from strings to numbers can be tricky in certain languages. Now, it's even trickier since the check digit of an ISBN-10 may be 'X' (representing '10'). For instance 3-598-21507-X is a valid ISBN-10.

Bonus tasks

- Generate a valid ISBN-13 from the input ISBN-10 (and maybe verify it again with a derived verifier).
- Generate valid ISBN, maybe even from a given starting ISBN.

36. Detect saddle points in a matrix. So say you have a matrix like so:

```
    1  2  3
   |---------
1 | 9  8  7
2 | 5  3  2     <--- saddle point at column 1, row 2, with value 5
3 | 6  6  7
```

It has a saddle point at column 1, row 2. It's called a "saddle point" because it is greater than or equal to every element in its row and less than or equal to every element in its column. A matrix may have zero or more saddle points. Your code should be able to provide the (possibly empty) list of all the saddle points for any given matrix. The matrix can have a different number of rows and columns (Non square). Note that you may find other definitions of matrix saddle points online, but the tests for this exercise follow the above unambiguous definition.