Perl 6 Intro Exercises

166 problems in three parts

Basic Exercises

64 problems

1. Input and printing

1.1. Sum of numbers

Write a program that takes three numbers and prints their sum. Every number is given on a separate line.

Hint: use `prompt()`.

1.2. Hello, Name!

Ask the user to enter their name. Write a program that greets the user by printing the word `Hello`, a comma, the name of the user and an exclamation mark after it.

Hint: use ~.

1.3. Apple sharing

N people take *K* apples and distribute them among each other evenly. The remaining (the indivisible) part is kept in the basket. How many apples each single person gets? How many apples remain in the basket?

The program reads the numbers *N* and *K*. It should print the two answers for the questions above.

1.4. Previous and next

Write a program that reads an integer number and prints its previous and next numbers.

1.5. Class desks

A school replaces the desks in three classrooms. At each desk, two students sit. Given the number of students in the class, print the smallest possible number of desks that can be purchased.

The program reads three integers (the numbers of students in each of the three classes) and prints an integer result.

Hint: use the modulo operation.

2. Integers and floating-point numbers

2.1. The last digit

Enter an integer value. Print its last digit.

Note: Solve this task in two variants: using numerical or string operations.

2.2. Tens

Given the number, print the digit corresponding to *tens*. E.g., for the number 234, print 3.

2.3. Sum of digits

Enter a three-digit number. Print the sum of its digits.

Note. Solve the task in three variants: 1) using numeric calculations, 2) using string operations, 3) using loops.

2.4. Fractional part

Given a positive real number, print its fractional part.

2.5. First digit after decimal point

Given a positive real number, print its first digit to the right of the decimal point.

2.6. The travelling car

A car covers the distance of *N* kilometers a day. How many days will it take to cover a route of *M* kilometers? The program gets two numbers: *N* and *M* and prints an integer result.

2.7. Digital clock

Given the integer number of minutes passed since midnight, tell how many hours and minutes are displayed on the 24-hour digital clock?

3. Control flow

3.1. Minimum

Given the two integer values, print the minimum one.

Note: Solve the task in two variants: with and without the `if` statement.

3.2. Sign of a number

For the given integer X, print 1 if it is positive, −1 if it's negative, or 0 if it equals to zero.

3.3. Maximum of three numbers

Given three integers, print the biggest value.

3.4. Equal numbers

Given three integers, determine how many of them are equal to each other.

3.5. The chess move

Given two different cells of the chessboard, determine whether a rook can go from the first cell to the second in one move.

The program receives the input of four numbers from 1 to 8, each specifying the column and row number. It prints either YES or NO.

Note: Use the below numbering of the chessboard cells. Solve the task for other chess pieces: king, queen, bishop, knight, and pawn.
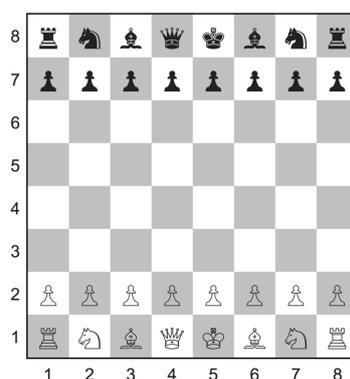
4. Strings

4.1. Parts of a word

Enter a long word. On separate lines, print the following parts of it:

1. The third character.
2. The second to last character.
3. The first five characters.
4. All but the last two characters.
5. All the characters with even indices (count 0 as even too).
6. All the characters with odd indices
7. All the characters in reverse order.
8. Every second character in reverse order, starting from the last one.

4.2. Length of a string

Enter a string. Print its length ignoring any potential spaces that you enter before or after the string.

4.3. Number of words

Given a string consisting of words separated by spaces. Determine how many words it has.

4.4. Cut the word

Given a word, cut it into two halves (or more or less equal parts), and print them in different order. E.g., `hello` becomes `llohe`.

4.5. Reverse a two-word phrase

Enter a phrase containing two words. Print the words in different order.

4.6. First and last occurrences

Enter a phrase. Enter a letter. Find the position of the first and the last occurrences of the given letter in the phrase.

4.7. The second occurrence

Find the second occurrence of the letter in the given phrase.

4.8. Replace the substring

In a given string, replace all occurrences of the number `1` with the word `one`.

4.9. Replace the substrings — 2

Expand the previous task to replace all the digits from 0 to 9.

Note. Limit the task to digits only, not numbers.

4.10. Delete a character

Enter a string. Remove all occurrences of the character `@` from it.

4.11. Replace with the fragment

In a given string, replace every occurrence of the letter *h* by the letter *H*, except for the first and the last ones.

4.12. Delete every third character

In a given string, delete all the characters whose indices are divisible by 3.

5. Loops, sequences, and ranges

5.1. Series

Given two integers A and B ($A \leqslant B$), print all numbers from A to B inclusively.

5.2. Reversed series

Update the previous task so that in the case $B \leqslant A$, the series is printed in the reversed order.

5.3. Sum of ten numbers

One at a line, ten numbers are given in the input. Read them and print their sum.

Note: Use as few variables as you can.

5.4. Sum of N numbers

Update the previous task so that it works for arbitrary number of input values. First, you should ask to enter the total number of data.

5.5. Sum of cubes

For the given integer N, calculate and print the result of the following formula:

$$1^3 + 2^3 + \ldots + N^3$$

5.6. The number or zeros

Given a few numbers: the first number in the input is N, after which N integers are given. Count the number of zeros among the given integers and print it.

Note: count the occurrences of zero numbers, not digits.

5.7. Factorial

Enter a number. Print its factorial.

Note 1. Factorial of N, or N!, is a product of all integer numbers between 1 and N.

Note 2. Solve the problem 1) without using functions; 2) using a recursive function.

5.8. Sum of factorials

Given an integer *N*, print the sum 1! + 2! + 3! + ... + *N*!

Note: Try not to use nested loops.

5.9. Stairs of digits

Enter an integer number. Print the corresponding number of lines so that the first line contains 1, and each consequent line contains one more digit.

E.g., for the number of 3 print the following:

```
1
12
123
```

5.10. Missing number

Enter four different numbers between 1 to 5 in random order. Print the missing number.

5.11. Least divisor

Given an integer greater or equal 2. Print its smallest integer divisor greater than 1.

5.12. The power of two

For a given integer *N*, find the greatest integer *x* where 2^x is less than or equal to *N*. Print the exponent value and the result of the expression 2^x.

5.13. The length of the sequence

Given a sequence of non-negative integers, where each number is written in a separate line. Determine the length of the sequence, where the sequence ends with 0 (more numbers follow). Print the length of the sequence. E.g., for the sequence `1 3 5 9 0 1 4`, the answer is 4.

5.14. The sum of the sequence

Determine the sum of all elements in the sequence, ending with the number 0.

5.15. The average of the sequence

Determine the average of all elements of the sequence ending with the number 0.

5.16. The maximum of the sequence

A sequence consists of integer numbers and ends with the number 0. Determine the largest element of the sequence.

5.17. The index of the maximum value

In the previous task, print also the index of the maximum number.

5.18. The number of elements that are greater than the previous one

A sequence consists of integer numbers and ends with the number 0. Determine how many elements of this sequence are greater than their neighbours above.

*5.19. The second maximum

The sequence consists of distinct positive integer numbers and ends with the number 0. Determine the value of the second largest element in this sequence. It is guaranteed that the sequence has at least two elements.

*5.20. The number of elements equal to the maximum

A sequence consists of integer numbers and ends with the number 0. Determine how many elements of this sequence are equal to its largest element.

Note. Do not store the values in a list.

*5.21. The maximum number of consecutive equal elements

Given a sequence of integer numbers ending with the number 0. Determine the length of the widest fragment where all the elements are equal to each other.

6. Arrays and Lists

6.1. Even indices

Given a list of numbers, find and print all the array elements with an even index number. (i.e., `@a[0]`, `@a[2]`, `@a[4]`, ...).

6.2. Even elements

Given a of numbers, find and print all elements that are an even number.

Note. Use a for-loop that iterates over the list, but not over its indices.

6.3. Greater than previous

Given a list of numbers, find and print all the elements that are greater than the previous element.

6.4. Neighbours of the same sign

Given a list of numbers, find and print the first adjacent elements which have the same sign. If there is no such pair, inform about that.

*6.5. Greater than neighbours

Given a list of numbers, determine and print the quantity of elements that are greater than both of their neighbours.

Note. Ignore the first and the last items of a list as they do not have enough neighbours.

6.6. The number of distinct elements

Given a list of numbers with all of its elements sorted in ascending order, determine and print the number of distinct elements in it.

6.7. Swap neighbours

Given a list of numbers, swap adjacent items in pairs (`A[0]` with `A[1]`, `A[2]` with `A[3]`, etc.). Print the resulting list. For the lists with odd number of elements, leave the last element in place.

6.8. Swap min and max

Given a list of unique numbers, swap the minimal and maximal elements of this list. Print the resulting list.

*6.9. Unique elements

Given a list of numbers, find and print the elements that appear in the list only once. The elements must be printed in the order in which they occur in the original list.

*6.10. Find the maximums in nested lists

Given a nested list of lists of integers, find the maximum values in each sub-list and create the list of maximums.

7. Bags, sets, and hashes

7.1. Intersection of sets

Given two lists of numbers. Find all the numbers that occur in both the first and the second list and print them in ascending order.

7.2. Seen before?

Given a sequence of numbers, determine if the next number has already been encountered. For each number, print YES if this number has already been encountered, and print NO otherwise.

*7.3. Dictionary of synonyms

You are given a dictionary of word pairs. Every word is a synonym to the other word in its pair. All the words in the dictionary are different.

For the given word, find its synonym.

*7.4. Countries and cities

Given a list of countries and cities of each country. Enter a city name. Print the country.

Extra exercises

58 problems

8. Strings

8.1. Remove blanks from a string

Remove leading, trailing, and double spaces from a given strings.

8.2. Camel Case

Create a camel-case identifier from a given phrase. E.g., you enter "Max span" and get `MaxSpan`.

8.3. Incrementing filenames

Generate a list of ten filenames like `file1.txt`, `file2.txt`, etc.

8.4. Random passwords

Generate a random string that can be used as a password.

Note: Use letters, digits, and special characters.

8.5. DNA-to-RNA transcription

Convert the given DNA sequence to a compliment RNA.

Note: A DNA string contains four letters: A, C, G, and T. An RNA string is built of A, C, G, and U. The transformation is shown below:

DNA	A	C	G	T
RNA	U	G	C	A

8.6. Plural endings

Put a noun in the correct form—singular or plural—depending on the number next to it.

8.7. The most frequent word

In the given text, find the most frequent word.

*8.8. The longest common substring

Find the longest common substring in the given two strings.

8.9. Anagram test

Tell if the two words are anagrams of each other.

8.10. Palindrome test

Check if the entered string is palindromic.

Test cases:

> *Never odd or even.*
>
> *Was it a rat I saw?*
>
> *Mr. Owl ate my metal worm.*

9. Numbers

9.1. π

Print the value of π.

9.2. Fibonacci numbers

Print the Nth Fibonacci number.

Note: Fibonacci numbers are defined by the formula:

$$f_n = f_{n-1} + f_{n-2}$$

Note 2. Assume the first two numbers being 0 and 1.

9.3. Approximate comparison

Compare two floating-point numbers approximately. Consider them equal if their relative values are not exceeding 1%.

Note. Test also negative numbers and zeros.

*9.4. Prime numbers

Decide if the given number is a prime number.

Note. Prime numbers are those that can be divided only by 1, and by themselves.

*9.5. List of prime numbers

Print the list of the first ten prime numbers.

9.6. Generate random numbers

Generate a random integer number between 0 and N.

9.7. Histogram of random numbers

Test the quality of the random generator by using a histogram to visualize the distribution. Write a program that prints the distribution and use Excel to draw the graph.

9.8. Standard deviation

For the given data, calculate the standard deviation value (sigma).

Note 1. Standard deviation is calculated using the formula:

$$\sigma = \sqrt{\frac{\Sigma(x_i - \bar{x})^2}{N - 1}}.$$

9.9. Polar coordinates

Convert the Cartesian coordinates to polar and backwards.

Note. Use the following formulae and implement the algorithm for positive x and y only.

$$x = r \cos \varphi, \qquad y = r \sin \varphi \,;$$

$$r = \sqrt{x^2 + y^2}, \qquad \varphi = \arctan \frac{y}{x}.$$

9.10. Monte Carlo method

Calculate the area of a circle and the volume of a sphere of radius 1 using the Monte Carlo method.

The Monte Carlo method is a statistical method of calculating data whose formula is not known. The idea is to generate a big number of random numbers and see how many of them satisfy the condition.

Note. Area of a circle is πr^2; volume of a sphere is $\frac{4}{3} \pi r^3$.

9.11. Compose the largest number

Given the list of integers, compose the largest possible number by concatenating them.

9.12. Convert to Roman numerals

Convert an integer number to a Roman numerals string.

9.13. Spelling numbers

Write a program that takes an integer below one million and prints its English spelling.

10. Aggregate data types

10.1. Reverse a list

Create a list out of five values. Create a reversed list and print it.

10.2. Rotate a list

Move all elements of a list N positions to the left. Restore the original list by rotating it to the right.

10.3. Randomise a list

Create a list with integer numbers from 1 to 20. Create another list that contains those numbers in random order. Print the result.

10.4. Incrementing list elements

Given a list of integers, increment each element of it and print the result.

10.5. Adding up two lists

Having two lists of the same length, create the third one, whose elements are the sums of the corresponding items of the initial lists.

10.6. Exclusion of two lists

From the given two lists, find the elements of the first list which do not appear in the second one.

10.7. Sum of the elements

Find the sum of the elements of an array of integers

10.8. Average of a list

Find the average value of the given list of numbers.

10.9. Moving average

Calculate the moving average for the given list of random numbers. Visualize it with Excel.

10.10. First odd number

Find the first odd number in a list of integers.

10.11. Every second element

Create a new list by picking every second element from the original list.

10.12. Increasing sequences

Check if the given list contains increasing (or decreasing) numbers.

10.13. Transpose a matrix

Take a matrix 2×2 and print its transposed version.

Note. Make it possible to expand the solution for bigger matrices.

10.14. Product table

Generate and print the product table for the values from 1 to 10.

10.15. Pascal triangle

Generate the numbers of the Pascal triangle and print them.

Note 1. The Pascal triangle is a sequence of rows of integers. It starts with a single 1 on the top row, and each following row has one number more, starting and ending with 1, while all of the other items are the sums of the two elements above it in the previous row. It is quite obvious from the illustration:

```
        1
       1 1
      1 2 1
     1 3 3 1
    1 4 6 4 1
   1 5 10 10 5 1
  1 6 15 20 25 6 1
```

Note 2. First, do not format the output. Then, try to make it look like in the example above.

11. Regexes

11.1. Count vowels

Count the number of vowel letters in the given word.

11.2. Count words

Count the number of words in a text.

11.3. Currency converter

Parse the string with a currency conversion request such as *'10 CHF in EUR'* and print the result.

11.4. Double each character

In a given string, double each alphanumeric character and print the result. Punctuation and spaces should stay untouched.

11.5. Remove duplicated words

Remove repeated words from a sentence.

*11.6. Separate digits and letters

In a given string that contains letters and digits, insert dashes on the borders between the digit and the letter sequences.

*11.7. Separate group of digits

Put commas between the three-digit groups in a big number, e.g.: 123,456,789.

11.8. Templating engine

Implement a simple templating engine that substitutes values in placeholders of the form `%name%`.

11.9. Highlight the duplicated words

Find repeated words and mark them with the `<b>` HTML tag.

Note: see also task E4.5.

11.10. Find the dates

In a given string, find all substrings that contain dates written in one of the known date format, e.g., `YYYY-MM-DD` and `DD/MM/YYYY`.

*11.11. Gray colours in CSS

For a given CSS file, find all the lines that contain a CSS selector representing gray colours.

Note: Colours are written in one of the following formats:

```
#ABABAB
#CCC
rgb(100, 100, 100)
rgba(100, 100, 100, 0)
hsl(0, 0%, 30%)
hsla(0, 0%, 30%, 0)
```

11.12. Markdown to HTML

Transform the Markdown text so that fragments between the stars become bold. Do not modify double stars.

An example of a text with Markdown formatting: `Hello, *World*!`

What you should get after the transformation: `Hello, <b>World</b>!`

11.13. Numbers

Find all integer numbers in a text, which may have spaces or commas as separators. E.g.: `12 345` or `12,345`, or `12345`.

11.14. IP addresses

Check if a given string is a valid IP address.

Note. Limit the solution to IPv4 only.

11.15. Base of a number

For a string that contains a number, tell if it is a valid decimal, or binary, or octal, or a hexadecimal number.

11.16. URL addresses

Tell if a given string is a valid address of a website (with an optional protocol scheme).

11.17. Markdown links

Convert all the links in a given Markdown text to HTML.

Note. The links in Markdown look like this: `[Wikipedia](https://wikipedia.org)`.

11.18. Sentences

Extract sentences from a given text and store them in a list.

11.19. Remove double spaces

Fix a given text by replacing all adjusted spaces by a single space.

11.20. Chemical elements

Tell if a given string contains a name of one of the known chemical elements.

Note 1. For simplicity, select a subset of 20-30 elements.

Note 2. Try optimizing the solution.

Rehearsal exercises

44 problems

12.1. Line equation

Enter coordinates of the two points on a surface. Print the equation of a straight line that crosses those points. Check the output if the calculated coefficients are negative.

12.2. Formatted output

Print four numbers as an aligned table 2×2. First row: 10, 235; second row: 1000, 50.

12.3. Properties of a cylinder

Enter three values: diameter and height of a cylinder and the area that you can paint with one can of paint. Calculate and print the number of cans you need to paint the cylinder.

12.4. Leap year

Given the distance from Sun and the speed of a planet, calculate and print the number of days in a year for that planet.

Note. Distance to Earth is 149.600.000 km. Earth orbit speed is 108.000 km/h.

12.5. Credit calculator

Calculate monthly payments for the credit given by the formula:

$$\frac{s \cdot p}{12} \cdot \frac{(1+p)^n}{(1+p)^n - 1},$$

where s is the total credit amount, p is the bank's interest rate, n is the credit term (in years). Also, print the total amount to be paid.

12.6. Repeated string

Repeat a given string the given number of times.

12.7. Properties of a triangle

Input the two sides of a right triangle and calculate its perimeter and area.

12.8. Swap values

Swap the values of two variables. Swap the values of three variables.

12.9. Random in range

Ask for two numbers and print a random number that lies in the given range. It should work for both integers and floating-point values.

12.10. Sum of digits

Enter an integer number of arbitrary length. Print the sum of its digits.

12.11. Properties of a number

Enter a number. Tell 1) if it is a one-digit or two-digit or more-than-three-digit number and 2) if it is positive or negative.

12.12. Interval function

Enter a number. Calculate the value of the function defined as:

$$f(x) = \begin{cases} x - 0.5 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ |x| & \text{if } x < 0 \end{cases}$$

Note. Find the ways to optimize it.

12.13. Square roots

Find the square roots of the quadratic equation:

$$ax^2 + bx + c = 0.$$

Note. Use the formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

12.14. Leap years

Tell if the given year is a leap year.

Note. Use the following algorithm:

- *if year is not divisible by 4 => common year*
- *if year is not divisible by 100 => leap year*
- *if year is not divisible by 400 => common year*
- *else => leap year*

12.15. If a triangle exists

Given the three numbers, tell if exists a triangle with such sides.

Note. Each side of a triangle must be longer than the sum of the other two.

12.16. Quadrant detection

Given the coordinates of a point on a surface, tell which quadrant it belongs to.

12.17. If is in circle

Determine if the given point *(x, y)* belongs to a circle of radius *r*.

12.18. Guess the number

One user enters a secret number. The second user tries to guess it: it enters a guess and gets a response: 'less' or 'more' until the number is correct.

12.19. Flip number

Flip the given integer number and print it.

12.20. Maximum powers

Given the number *N* and the power index *P*, print all the integer numbers, whose power *P* does not exceed *N*.

12.21. Function table

Print the table of sin (x) for x between 0 and 5. Use Excel to draw the graph.

12.22. ASCII table

Print the list of ASCII characters.

12.23. Maximum number

For the given input of ten integer numbers, find the maximum one without using arrays or the `max` function.

12.24. Distance between the points

Enter coordinates of two points on a surface. Find the distance between them.

12.25. Distance — 2

Solve the previous task using complex numbers.

12.26. Is unique

Tell if the given list of ten integers contains only unique numbers.

12.27. Filter odd and even

Enter ten numbers. Create a list where all odd numbers go first, and all even numbers are at the end of the list. Preserve the order of the entered data.

12.28. Replaces spaces with stars

Take a string. Replace all the spaces with *. If there are more than one space together, put a single star.

12.29. Find numbers

Take a string. Find and print all integer numbers that it contains.

12.30. Replace with brackets

In a given string, remove the substring that is located between a pair of square brackets. Try different inputs with nested brackets.

12.31. Shorten long words

In a sentence, replace long words with their first five characters followed by ellipsis.

12.32. Print N^{th} word

Print the word from the sentence by its number.

12.33. Reverse sentence

Take a sentence. Reverse the order of the words in it and print the new string.

Note: ignore punctuation for simplicity.

12.34. Count words

Count words in a file.

Note. Pass the filename via command line.

12.35. Loop the list

For a given list of integer values, shift all the elements two positions towards the end of the list, so that ending elements go to the beginning.

12.36. Count unique words

Enter a sentence. Count the number of different words in it.

13. Recursion

13.1. Factorial

Create a function for calculating factorial of an integer number.

13.2. Sum of integers

Create a function that calculates a sum of a list of integer numbers.

13.3. Power

Create a function to calculate N^{th} power of an integer value.

13.4. Integer sequence

Print all the integer numbers from 1 to 10.

13.5. Maximum

Find the maximum value in a list of integer values.

*13.6. Second maximum

Find the second maximum from a list of integer values.

13.7. Sum of digits

Find the sum of digits of a given number.

13.8. Prime numbers

Print the list of the first 10 prime numbers.