

02393 C++ Programming Exercises

Assignment 9

To be handed in via CodeJudge

<https://dtu.codejudge.net/02393-e19/assignments/>

The goal of the following exercises is for you to familiarize with recursion. All the exercises can be solved without recursion, but we suggest that you try to use recursion. In all cases input is to be read from `cin` and the result is to be provided to `cout`.

Reverse Write a program that reverses a sequence of integers, as provided in the standard input.

For example, if the input is

```
1 2 3 4 5
```

then the output should be

```
5 4 3 2 1
```

Fibonacci Write a program that computes the Fibonacci numbers F_{n_i} for a sequence n_0, \dots, n_k given in the standard input. See http://en.wikipedia.org/wiki/Fibonacci_number

For example if the input is

```
0 5
```

then the output should be

```
1 8
```

since F_0 is 1 and F_5 is 8.

The above page explains that there exist different interpretations of the Fibonacci sequence. In this assignment we consider the following:

$F_0 = 1, F_1 = 1, F_2 = 2, F_3 = 3, F_4 = 5, F_5 = 8, \dots$

Palindrome Write a program that decides whether a sequence of integers is a palindrome, i.e. if reading the sequence from right to left results in the very same sequence.

For example, if the input is

13 22 33 22 13

then the output should be

yes

but if the input is

13 22 31

then the output should be

no

since the right-to-left reading is

31 22 13

Note that the right-to-left reading does not refer to individual digits, but to entire numbers.

The Levenshtein distance The Levenshtein distance between two sequences of characters $u = u_1, u_2, \dots, u_k$ and $v = v_1, v_2, \dots, v_l$ is defined by:

$$d(u, v) = \begin{cases} |v| & \text{if } |u| = 0, \\ |u| & \text{if } |v| = 0, \\ \min \begin{cases} d(u^1, v) + 1 \\ d(u, v^1) + 1 \\ d(u^1, v^1) + f(u_1, v_1) \end{cases} & \text{otherwise.} \end{cases}$$

where:

- $|w|$ denotes the length of a sequence w ;
- w_1 denotes the first element of a sequence $w = w_1, w_2, w_3, \dots$;
- w^1 denotes the suffix w_2, w_3, \dots of a sequence $w = w_1, w_2, w_3, \dots$.
That is, w^1 corresponds to w without the first element (which is w_1);
- and, for e and e' being two characters, $f(e, e')$ is 0 when e and e' are the same character (i.e. $e = e'$), and 1 otherwise.

As an example you can easily check that the distance $d("AB", "B")$ between "AB" and "B" is 1 since:

$$\begin{aligned} d("AB", "B") &= \min(d("B", "B") + 1, d("AB", "") + 1, d("B", "") + 1) = \min(1, 3, 2) = 1 \\ d("B", "B") &= \min(d("", "B") + 1, d("B", "") + 1, d("", "") + 0) = \min(2, 2, 0) = 0 \\ d("AB", "") &= 2 \\ d("B", "") &= 1 \\ d("", "B") &= 1 \end{aligned}$$

Write a program that reads two words and returns their distance.

Challenges. Try to write a function that, given a set of elements, computes all the possible permutations of the elements. Try to write a function that, given a set, computes its powerset.