

## 08. Complex tasks

**8.01.** Tisho wants to play *Bulls and Cows*, but he is confused by the rules, so he needs our help. We have to write a program that reads two integer numbers A and B in the range [0,9999] from the console and prints back the number of bulls and cows for these numbers. Numbers smaller than 1000 we should pad with zeros, for example 19 should be interpreted as 0019.

| Example input | Expected output |
|---------------|-----------------|
| 1001 2010     | 1 2             |
| 0 1           | 3 0             |
| 9988          | 0 0             |
| 1010          | 2 2             |

**8.02.** Tisho really enjoyed our help and he needs us again. Now he asked us to do the following: Given a number A - Tisho's suggestion for his opponent's number, *b* - number of bulls and *c* - number of cows, we have to print every number *N* that has *b* bulls and *c* cows with the number A.

**8.03.** Strahil and Pesho really love hamburgers and they have a game which they play in rounds - they try to eat as much as they can. And since it is very hard to overeat, they can't remember how many burgers they ate, so they write the result in a single number B with the following property: the sum of the left digits of B gives the number of burgers Strahil ate, and the sum of the right digits gives the number of burgers Pesho ate. For example 3489 means that Strahil ate  $3 + 4 = 7$  and Pesho ate  $8 + 9 = 17$ . If the number B has odd number of digits, then the one in the middle is shared between them both. For example 324 means  $3 + 2 = 5$  for Strahil and  $2 + 4 = 6$  for Pesho. The number B will never be more than 9 digits long. Your task is to help them decide who is better and print who won the game and how many burgers he ate. If they have equal results, print that the result is even.

| Example input | Expected output |
|---------------|-----------------|
| 1221          | Even            |
| 23            | Pesho 3         |
| 6543          | Strahil 11      |
| 489           | Pesho 17        |

**8.04.** Check if a given Sudoku board configuration is valid or not.

Example input:

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | - | - | 7 | - | - | - | - |
| 6 | - | - | 1 | 9 | 5 | - | - | - |
| - | 9 | 8 | - | - | - | - | 6 | - |
| 8 | - | - | - | 6 | - | - | - | 3 |
| 4 | - | - | 8 | - | 3 | - | - | 1 |
| 7 | - | - | - | 2 | - | - | - | 6 |
| - | 6 | - | - | - | - | 2 | 8 | - |
| - | - | - | 4 | 1 | 9 | - | - | 5 |
| - | - | - | - | 8 | - | - | 7 | 9 |

Expected output: true

Examples input:

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 4 | 6 | 7 | 8 | 9 | 1 | 2 |
| 6 | 7 | 2 | 1 | 9 | 5 | 3 | 4 | 8 |
| 1 | 9 | 8 | 3 | 4 | 2 | 5 | 6 | 7 |
| 8 | 5 | 9 | - | 6 | - | - | - | 3 |
| 4 | 2 | 6 | 8 | - | 3 | - | - | 1 |
| 7 | 1 | 3 | 9 | 2 | - | - | - | 6 |
| - | 6 | - | 5 | 5 | - | 2 | 8 | - |
| - | - | - | 4 | 1 | 9 | 6 | 3 | 5 |
| 3 | 4 | 5 | 2 | 8 | 6 | 1 | 7 | 9 |

Expected output: False

**8.05.** Write a program which by a given random text, displays all the different words in the text one by one in a row arranged lexicographically, starting from the smallest one. Use the standard character order with respect to the ASCII table in the lexicographic ordinance, considering that capital letters are before lowercase letters. For a word we take every string containing only lowercase and uppercase symbols from the Latin alphabet. Consider all other characters as separators between the words. Implement the code of the program without using libraries other than `<iostream>`. Each memory the program uses must be dynamically allocated to the exact size and correctly released.

[github.com/andy489](https://github.com/andy489)