

beCP

2022

Task 2.1: Secret Agents (secretagents)

Author: Damien Leroy

Time limit: 2s Memory limit: 64 MB

Note: This task is interactive. Please look at the special instructions to implement, compile and test your program. Don't hesitate to ask the staff for help.

Your secret agents have now been deployed on the war field. In order to announce their presence to others, they use a communication system which broadcasts their agent id through radio waves. In order to prevent any risk of tracking, the emitter may introduce errors in the id it sends.

Your task is to write the code of the encoder which maps the agent names to a 6-digit id, and of the decoder which maps a (possibly altered) id to its agent name.

Functions to implement

C++	<code>vector<int> encode(vector<string> agents)</code>
return	Given a list of length N with agent names (all different), generates a list of agent ids. This function is called once. A list of length N of agent ids (6-digit number). The id at index i corresponds to the agent <code>agents[i]</code> .

C++	<code>string decode(vector<string> agents, int id)</code>
return	Given a list of length N with agent names in same order as <code>encode</code> and an id, returns the agent name corresponding to the given id (possibly altered). The id always corresponds to an existing agent (with a possible alteration). This function is called up to 60000 times, always after <code>encode</code> . The agent name matching the given id.

General limits

- $2 \leq N \leq 1000$

Additional constraints

Subtask	Points	Constraints
A	40	$N \leq 100$, <i>ids</i> to be decoded are not altered
B	40	$N \leq 100$, <i>ids</i> to be decoded has, at most, one of its digits changed
C	20	<i>ids</i> to be decoded has, at most, one of its digits changed

Technical details

In order to test locally, we provide you with a grader (different from the one on the server) that you can modify freely. This grader does **not** have to be submitted. You can compile locally your code with the grader using:

```
g++ -std=c++11 -Wall -Werror -Wshadow grader.cpp secretagents.cpp
```

Example 1 (subtask A: ids are not altered)

Suppose `encode` is called with the following `agents` list:

Rohaam Wilkerson	Anwar Tierney	Caroline Scott
------------------	---------------	----------------

You may return, for instance:

1 (= 000001)	900990	249234
--------------	--------	--------

Then, the `decode` function may be called several times, with the same `agents` as above and the following `ids`:

id	expected answer
900990	Anwar Tierney
1	Rohaam Wilkerson

Example 2 (subtask B/C: ids may be altered)

Suppose `encode` is called with the following `agents` list:

Murphy Aguirre	Kierran Heath	Anton Russo	Claudia Rogers
----------------	---------------	-------------	----------------

You may return, for instance:

123456	990000	30 (= 000030)	666666
--------	--------	---------------	--------

Then, the `decode` function may be called several times, with the same `agents` as above and the following `ids`:

id	expected answer
990400	Kierran Heath
800030	Anton Russo
0 (= 000000)	Anton Russo
666666	Claudia Rogers

Hints: Extracting digits from a number

In order to extract one specific digit from a number, you have to use the integer division (/) and the modulo (%) operators. The integer division is a division in which the fractional part (remainder) is discarded, for instance $1234/100$ returns the integer 12. The modulo returns the remainder of the division, so $1234\%100$ returns 34.

Combining these two allows you to get one specific digit from a number, for instance $(1234/100)\%10$ returns 2, $(1234/10)\%10$ returns 3.

In the same way, you can build numbers from digits using multiplication. If you have to repeat 12 twice in a number, you just have to do $12 * 100 + 12$. If you want to add 5 in front, just add $5 * 10000$ to the number, etc.