Password dimenticata (account)

Davide is training hard for OII (Italian Olympiad in Informatics) by solving problems on the training platform `training.olinfo.it`.

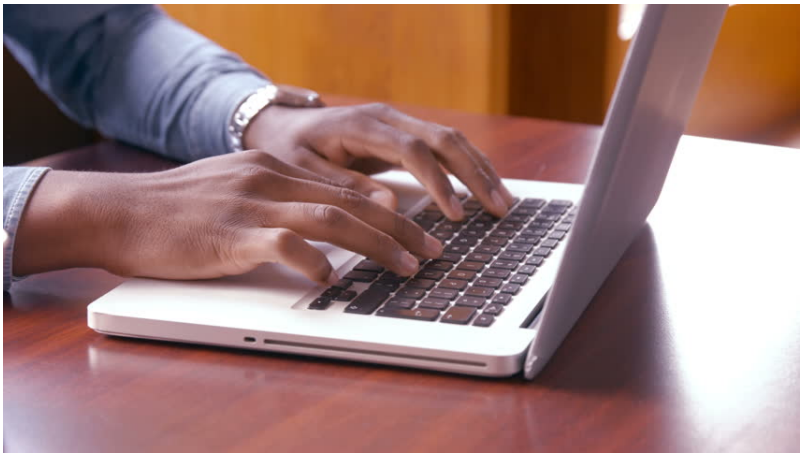

Figure 1: Davide typing 150 words per minute

Davide is training so hard that he has forgotten some very important things, including his password! He remembers that his password contains only the first N characters of the lowercase English alphabet and each character appears exactly once. Luckily, `training.olinfo.it` has a powerful password recovery tool[1]. The tool, given a number K, returns the length of the K-th contiguous substring of the password in lexicographical order.

Help Davide recover his password!

Implementation

You should submit a single file, with either a `.c` or `.cpp` extension.

> ☞ Among the attachments of this task you will find a template `account.c` or `account.cpp` with a sample implementation.

You will have to implement the following function:

Function `indovina`

C/C++	`void indovina(int N);`

The function `indovina` is called (once, at the beginning) with the following values of the parameters:
- The integer N which represents the password length.

Your program can call the following functions, which are already defined in the grader:

Function `chiedi`

C/C++	`int chiedi(int K);`

The function `chiedi` may be called (multiple times) with the following parameters:
- The integer K which represents the index of the substring **starting from 1**.

[1] Unfortunately the tool is no longer available due to security concerns.

– The function returns the length of the K-th substring in increasing order.

■ Function `rispondi`

C	`void rispondi(const char* S);`
C++	`void rispondi(string S);`

The function `rispondi` must be called (once) at the end, with the following parameters:
– The string S that represents Davide's password.

Sample Grader

Among this task's attachments you will find a simplified version of the grader used during the evaluation, which you can use to test your solutions locally. The sample grader reads data from `stdin`, calls the functions that you should implement and writes back on `stdout` using the following format.

The input file is formed by 2 lines, containing:

- Line 1: the integer N.
- Line 2: the string S.

The output file is formed by a single line, containing "`Wrong answer`" if the password is wrong, "`Correct answer`" followed by the number of calls to `chiedi` otherwise.

Constraints

- $1 \leq N \leq 26$.

Scoring

Your program will be tested against several test cases grouped in subtasks. The score in each subtask will be calculated as the minimum score obtained in any of its test cases, multiplied by the value of the subtask. For every subtask except the last, the score of a test case is 1 if the password is correct, 0 otherwise.

For the last subtask, the score of a test case is 0 if the password is wrong, the value of the following formula otherwise:

$$\min\left\{\sqrt{\frac{N}{Q+1}}, 1\right\}$$

where Q is the number of calls to the function `chiedi`.

- **Subtask 1 [0 points]**: Examples.
- **Subtask 2 [10 points]**: $N \leq 3$
- **Subtask 3 [15 points]**: $N \leq 8$
- **Subtask 4 [10 points]**: $N \leq 10$
- **Subtask 5 [15 points]**: $N \leq 16$
- **Subtask 6 [50 points]**: $N \leq 26$

Examples

stdin	stdout
5 caedb	Correct answer: 4 calls

Explanations

The substrings of `caedb` in lexicographical order are:

- `a`
- `ae`
- `aed`
- `aedb`
- `b`
- `c`
- `ca`
- `cae`
- `caed`
- `caedb`
- `d`
- `db`
- `e`
- `ed`
- `edb`

A possible execution might be:

- `chiedi(1)`: the first substring is `a`, the function returns 1.
- `chiedi(15)`: the fifteenth substring is `edb`, the function returns 3.
- `chiedi(9)`: the ninth substring is `caed`, the function returns 4.
- `chiedi(10)`: the tenth substring is `caedb`, the function returns 5.
- `rispondi("caedb")`: the answer is correct.