Password

I forgot my password again! I am sitting at my computer punching in wrong passwords. All I remember is that my password contains only lowercase letters. Luckily, the login system responds with more than just "wrong password". It also tells me the length of the longest prefix of my input that occurs as a (not necessarily contiguous) subsequence in the password. Formally, for a password $P = p_1 p_2 \ldots p_N$ and input $Q = q_1 q_2 \ldots q_N$, the login system's answer is the largest L for which there exist indices $1 \le k_1 < k_2 < \ldots < k_L \le N$ such that $q_i = p_{k_i}$ for all $1 \le i \le L$. The system also tells me N, the length of my password, and S, meaning my password only uses the first S letters of the alphabet. For example, $S = 4$ means my password only contains *a*, *b*, *c* and *d* (but not necessarily all of them).

Please help me recover my password!

Implementation details

This problem is interactive. You should implement the function

```
C:    char* guess(int n, int s);
C++:  string guess(int n, int s);
```

- Arguments: N and S as described above.
- Return value: the correct password.

Your program may call the function

```
C:    int query(char* str);
C++:  int query(string str);
```

- Argument: a string of 1 to N letters from among the first S letters of the alphabet.
- Return value: an integer between 0 and the length of *str*, representing the login system's answer to your query.
- To avoid compiler errors, **you should declare this function using the exact text above**, anywhere in the global scope before calling it.
- You may call this function **at most 50,000 times** for each test case.

Your program may define additional functions.

Sample grader

A sample `grader.{c,cpp}` is provided for you to test your code locally. The sample grader reads the input from the file `password.in` in the format:

- line 1: *N S*
- line 2: password

You can compile the grader together with your solution. Then you can run the resulting binary to test your guessing strategy against the given input.

Limits

- Time limit: 3.0 seconds
- Memory limit: 512 MB

Subtasks

Test cases may be **scored as a group**. In order, the subtasks are:

Subtask	Percentage of points	Input constraints
1	10%	$N \le S \le 26$; all the letters in the password are distinct
2	20%	$2 \le N \le 100$ and $2 \le S \le 4$
3	20%	$2 \le N \le 2{,}000$ and $2 \le S \le 20$
4	30%	$2 \le N \le 3{,}500$
5	20%	$2 \le N \le 5{,}000$

Example

Let the password be *aab*. The grader calls `guess(3, 2)`. The call log may be

Call	Return value
`guess("ab")`	`2`
`guess("abb")`	`2`
`guess("bab")`	`1`
`guess("aab")`	`3`

At this point, `guess(3, 2)` should return `"aab"`.