Competitive Programming SS24

Submit until end of contest

Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Problem: Cracking Passwords (2 second timelimit)

Any similarity to actual persons living or dead is purely coincidental.

You've just started your new job as a Computer Programming (CompProg) tutor and have been assigned the task of setting up the upcoming homework contest. Simple enough, you think.

You start up the web browser, open the DOMJudge interface, fill in your username, fill in your password. Fill in your password. You stare at the blank field. What was it again?

You try all your usual passwords like "hello", "password", and "1234", but none of them work. You must have picked a secure one this time! You knew it was a bad idea to follow the advice of cybersecurity "experts".

Sure, you could ask your fellow tutors to reset it, but how embarrassing would that be? Besides, it's not the first time you've forgotten your password.

Determined to solve this on your own, you start digging through DOMJudge's source code. You discover that the system accepts passwords between 1 and 20 characters, consisting of uppercase or lowercase English letters and digits. Even better, you manage to find the source code for the password checking mechanism, along with execution times for various operations.

```
bool check_password(string guess, string password) {
    if (guess.size() != password.size())
        return false;
    for (int i = 0; i < guess.size(); i++)
        if (guess[i] != password[i])
            return false;
    return true;
}
```

Figure F.1: The C++ code for checking the password.

- A branching statement (if or for) takes 1 ms.
- An assignment, or update of a memory address takes 1 ms.
- A comparison between two memory addresses takes 3 ms.
- A return statement takes 1 ms.

Armed with this knowledge, can you recover your password?

Interaction This is an interactive problem. Your submission will be run against an *interactor*, which reads the standard output of your submission and writes to the standard input of your submission. This interaction needs to follow a specific protocol:

- Your program first sends a password string, consisting of between 1 and 20 English lowercase, uppercase letters, or digits.
- Depending on if the password is correct, the interactor then responds with either:
  - If the password is correct; "`ACCESS GRANTED`". Your program should then exit cleanly.
  - If the password is incorrect; "`ACCESS DENIED (`t `ms)`", where t is the time it took to verify the password in ms. Your program can then make another guess.

Make sure you flush the buffer after each write. You can guess at most 2 500 times. A testing tool is provided to help you develop your solution.

Pro tip: You can use

- `getline(cin, line)` and `stoi(line.substr("ACCESS␣DENIED␣("s.size()))` in C++
- `line = input()` and `int("".join(c for c in line if c.isdigit()))` in Python

to extract the time from the response.

Read	Sample Interaction 1	Write
	`A`	
`ACCESS DENIED (5 ms)`		
	`HunFhun`	
`ACCESS DENIED (41 ms)`		
	`Hunter1`	
`ACCESS DENIED (68 ms)`		
	`Hunter2`	
`ACCESS GRANTED`		