Debug E

Download DebugE.cpp

Lab E

Problem

Passwords were once thought to be a thing of the past, but they are still widely used in today's society. However, places that use passwords are now starting to require very complex passwords that nobody can remember, hence people write them down and defeat any purpose for complexity. Some may argue we're reaching a point where the passwords are so complex, it causes people to do unsafe things with their password, such as emailing it to themselves.

You are designing a password-generating program that will use the random number generator and a defined array to produce characters for the user. The user will supply how long they want their password, and then either accept or reject the passwords that your program generates.

NOTE: DO NOT use this program to generate actual passwords. The random number generator you will be using isn't considered "cryptographically" secure for good reasons!

Inputs

You will need to ask the user how long they want their password to be (up to a maximum). After you generate a password, you will need to ask the user if they accept or reject the password. If they reject it, generate another password with the desired size and repeat the process until the user accepts the password.

NOTE: Since Code Assessor will be grading your work, the seed given to your program and Code Assessor needs to be the same. For grading purposes, you will need to input a seed from the user and use it to seed your random number generator.

Use the following prompts for your inputs:

> **"Enter seed: "**
> **"Enter password length: "**
> **"Accept password? "**

Process

Construct a global, constant array of characters that contains 10 elements. **Furthermore, use a global constant to store the size of the array.** Initialize the array to the following values:

```
Array Element:  0    1    2    3    4    5    6    7    8    9
Array Value  : 'a', 'b', 'c', 'd', 'E', 'F', '!', '*', '6', '2'
```

You must error check the desired password length. Should the user specify anything smaller than 1, set the length to 1. Should the user specify anything bigger than 25, set the length to 25.

You must seed the random number generator using the seed given by the user, otherwise you and Code Assessor will never get the same values. You will only seed your random number generator ONCE.

Create a new function that:

1. Generates a random number between 0 and SIZE-1 (SIZE being the size of the characters array)
    - The random number is the index into the characters array
2. Returns the character in the character array given by the random index

Outputs

Use your character generating function to generate the amount of characters specified by the user (after error checking, of course). After you present to the user their new password, output a blank, empty line. Then ask the user whether or not they accept the password. Regenerate another password of the same length until the user types either **y** or **Y** (as in, yes, accept the password).

Additional Requirements

1. You must create a function that generates a single character from the random number generator.
    - This function must be prototyped above int main and defined below int main
2. You must define a global constant for the character array and size and use these constants where appropriate.
3. You must use a global, constant character array to determine the characters of the password.
4. You must use an arithmetic operator to squeeze the random number between 0 and SIZE-1 (no loopes, etc).

C++ Topics Covered

Pre-defined functions
User-defined functions
Random number generator
Arrays

Textbook Chapters Covered

Chapter 4.1
Chapter 4.2 (Pre-defined functions)
Chapter 4.3 (User-defined functions)

Relevant Reading

Predefined functions
Random numbers
User defined functions