Hands-on Experiment # 5 : Worksheet

Section_____1_________ Date______ 24 February 2020_______________

No more than 3 students per one submission of this worksheet.

Student ID ____________Name___________________

Student ID ____________Name___________________

Student ID ____________Name___________________

Part A: Complex Conditions

An online service lets its users register for a new account by themselves. Each user can choose a username as well as a password. Here are the rules on picking a username and setting the associated password.

Username

- A username must be at least 4 characters in its length but must not exceed 255 characters.
- The first character of a username must be English alphabets. It can be either a lowercase letter (a-z) or an uppercase letter (A-Z).
- No spaces are allowed in a username.
- No backslashes are allowed in a username.
- Dots (.) can be used but no dots can be adjacent to one another.

Password

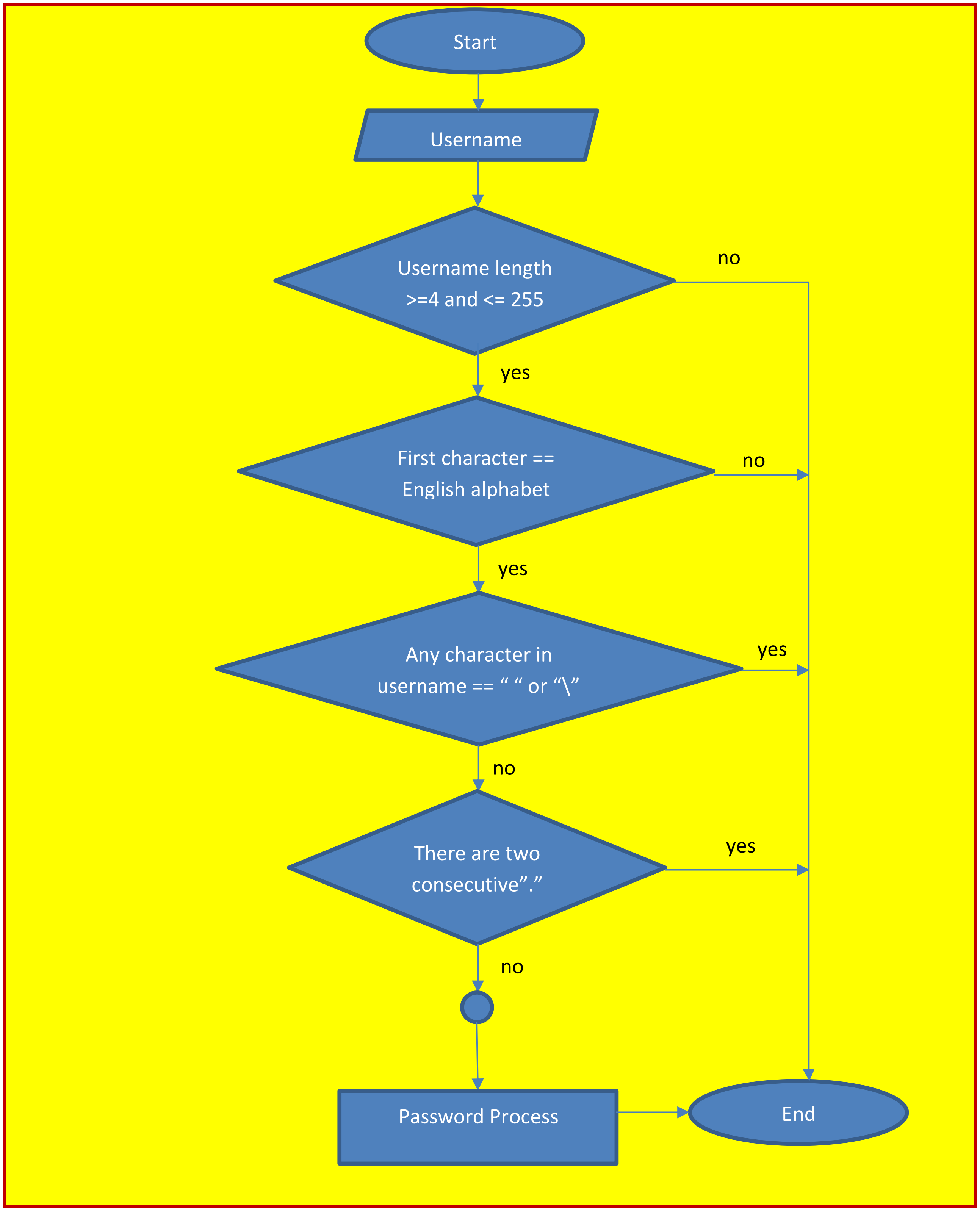
- A password must be at least 4 characters in its length but must not exceed 255 characters.
- A password must not contain the username regardless of their cases. This means that when checking whether the username is in the password, lowercase and uppercase letters are considered the same if they are the same alphabets. (E.g. A password is not allowed to contained “JaSoN” if the username is “jason”)
- If the length of a password is less than 8 characters, the password must contain at least 2 digits (0-9) and at least one of the following symbols: ! @ # $ %.
- An asterisk (*) cannot be used in a password.
- No spaces are allowed in a password.
- No backslashes are allowed in a password.

Determine whether the username/password combination in each row of the following table conforms to the rules above. If not, state the reason why.

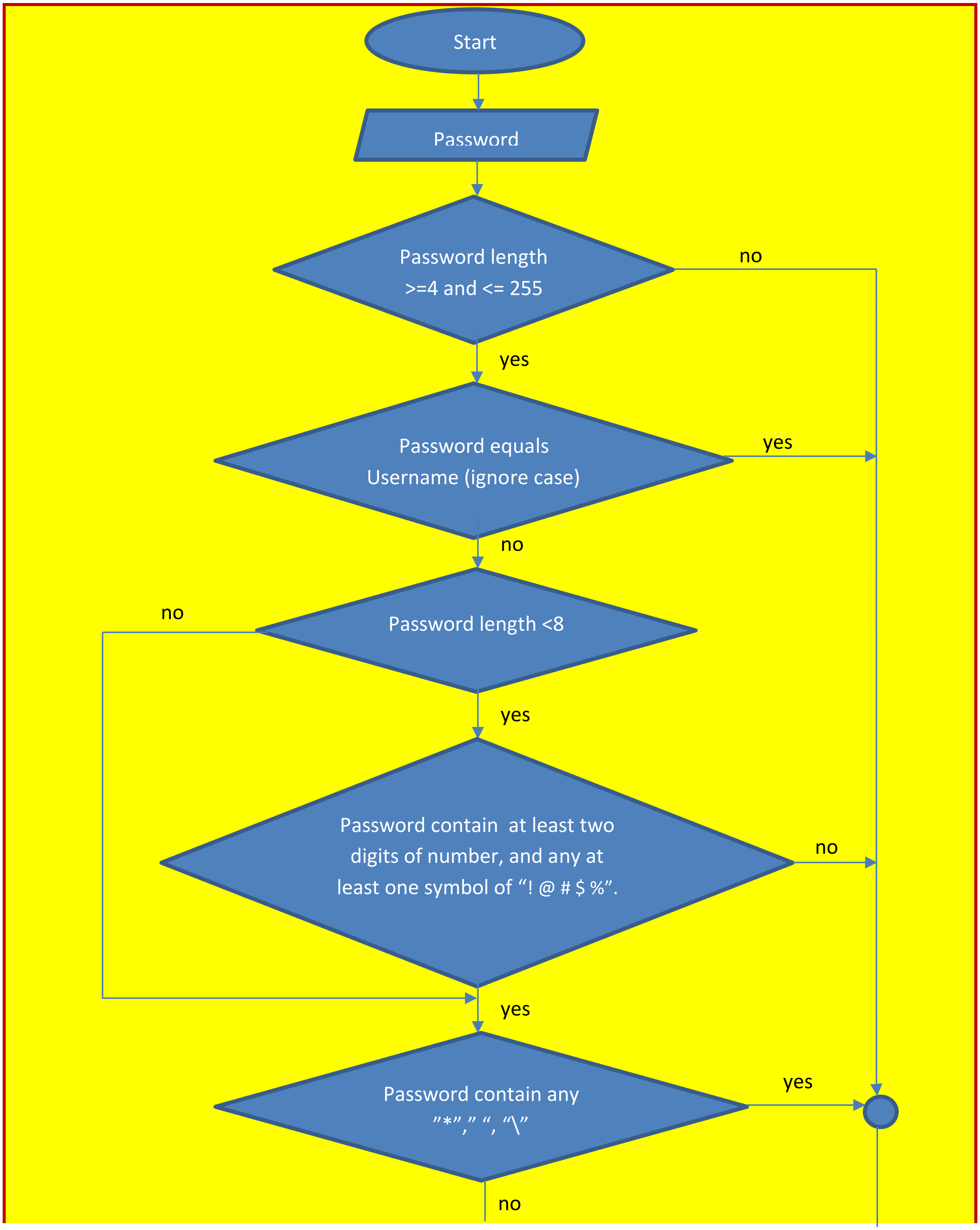
Username	Password	Conform to rules (Y/N)	If not, why?
Broccoli_Duke	5%x2	Yes	
Anna	itisANNA	No	Password contains username.
Anna.Stone.2	BGiskj129Dk8960	Yes	

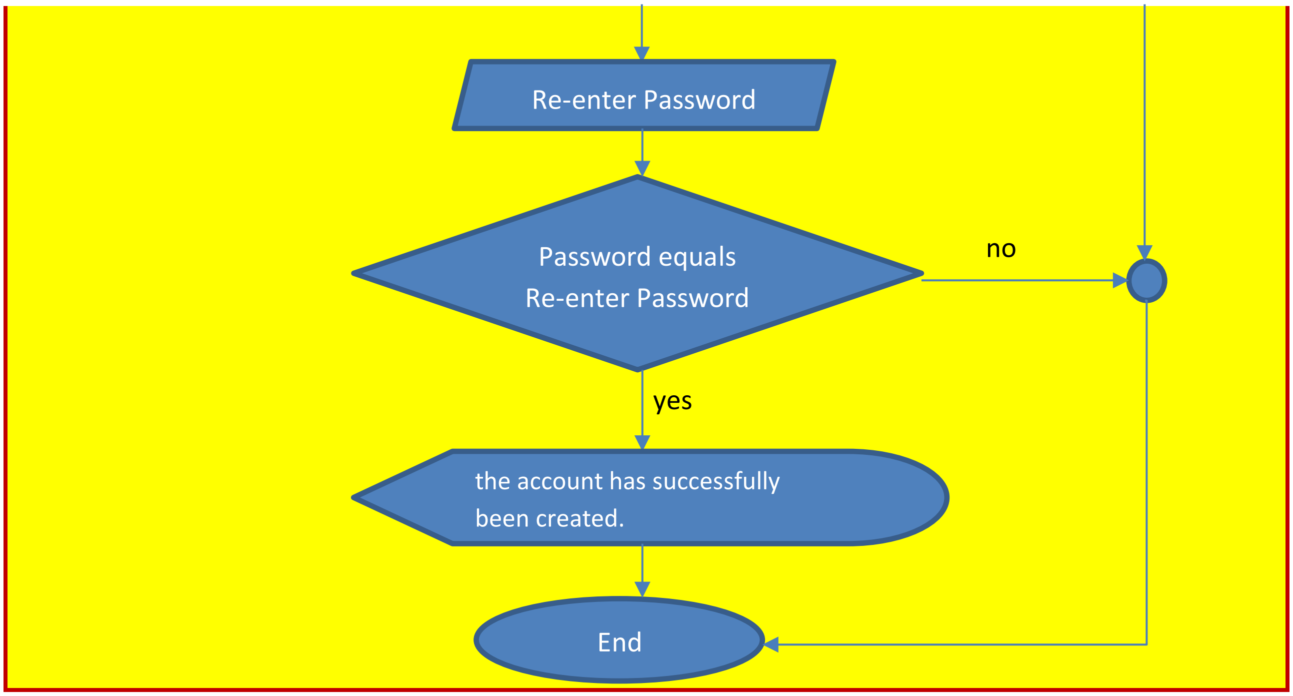
Anna……Stone	qwertyUIDFO0001_3	No	There are adjacent dots.
Anna Stone	ABCDEFGHIJK	No	There is a space in username.
5640021321	1234567890	No	First character is not a letter.
King.of.StarCraft	zerg	No	Password doesn’t contain two numbers and one symbol.
King.of.StarCraft2	********	No	Password contains *.
King.of.StarCraft3	09638	No	Password doesn’t contain at least one symbol.
Boxster999	0000 0000	No	Password contains a space.
Crafty\Random	Complicated_password	No	Username contains backslashes.

Draw a flowchart showing the process to be performed for verifying whether *an input username* conforms to the rules.



Draw a flowchart showing the process to be performed for verifying whether *an input username/password combination* conforms to the rules. Assume that the username has already been set so that it conforms to the rules.





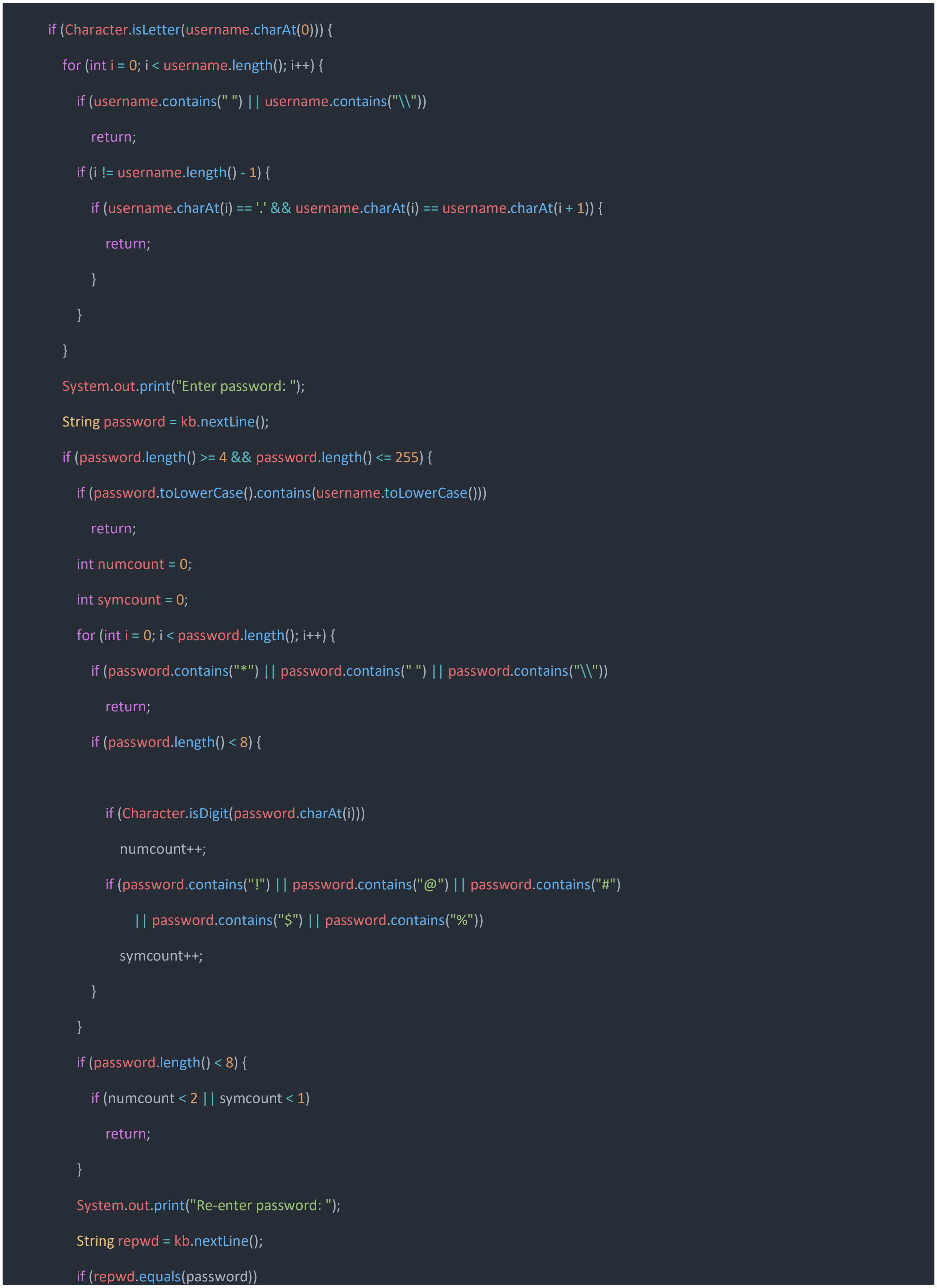
Write a Java program that performs the following steps.

1. Ask the user to set a username.
    a. If the username conforms to the rules, proceed to password setting.
    b. If the username does not conform to the rules, the program ends.
2. Ask the user to set a password.
    a. If the password conforms to the rules, proceed to password confirmation.
    b. If the password does not conform to the rules, the program ends.
3. Ask the user to re-enter the password for confirmation.
    a. If the password is the same as the one in 2, show a message saying that the account has successfully been created.
    b. If the password is different from the one in 2, the program ends.

Methods in the String class should be useful in this program. Check its API specification at http://docs.oracle.com/javase/7/docs/api/java/lang/String.html

List your source code below.

```
import java.util.Scanner;


public class PartA {

    public static void main(String[] args) {

        Scanner kb = new Scanner(System.in);

        System.out.print("Enter username: ");

        String username = kb.nextLine();

        if (username.length() >= 4 && username.length() <= 255) {
```



```
if (Character.isLetter(username.charAt(0))) {
  for (int i = 0; i < username.length(); i++) {
    if (username.contains(" ") || username.contains("\\"))
      return;
    if (i != username.length() - 1) {
      if (username.charAt(i) == '.' && username.charAt(i) == username.charAt(i + 1)) {
        return;
      }
    }
  }
  System.out.print("Enter password: ");
  String password = kb.nextLine();
  if (password.length() >= 4 && password.length() <= 255) {
    if (password.toLowerCase().contains(username.toLowerCase()))
      return;
    int numcount = 0;
    int symcount = 0;
    for (int i = 0; i < password.length(); i++) {
      if (password.contains("*") || password.contains(" ") || password.contains("\\"))
        return;
      if (password.length() < 8) {

        if (Character.isDigit(password.charAt(i)))
          numcount++;
        if (password.contains("!") || password.contains("@") || password.contains("#")
            || password.contains("$") || password.contains("%"))
          symcount++;
      }
    }
    if (password.length() < 8) {
      if (numcount < 2 || symcount < 1)
        return;
    }
    System.out.print("Re-enter password: ");
    String repwd = kb.nextLine();
    if (repwd.equals(password))
```

```
            System.out.print("Account has successfully been created.");
          else
            return;
        }
      }
    }
  }
}
```

Test the program with the username/password combinations listed in the first table.

Does your program verify those combinations correctly?

Yes

Submit this worksheet (by only one member of the group) via http://www.myCourseVille.com (Assignments > Hands-on Experiment # 5) before noon of the day after your lecture.