Password Generator [contributed by F. R. Salvador]

The rules for my.dlsu.edu.ph account passwords are copy/pasted below:

Password should comply with the following rules:

- It should not contain a part of the user name and full name.
- It should be at least eight (8) and at most fifty (50) characters long. It is best for the password to be at least 15 characters long.
- It should contain upper-case and lower-case letters.
- It must have at least one number.
- It must have at least one special character. Special characters are symbols on the keyboard which are not defined as letters or numerals. These are ` ~ ! ^ * ( ) _ - = { } | [ ] : ; ' < > ? , .
- It must not contain spaces.

YOUR TASK: Write a C program that will automatically generate a default password given the user's Firstname and Lastname as inputs. Both names are strings made up of lower case letters with no space in between. Follow the 5-step password generation algorithm given in the table below. Three examples are provided to help you figure out what is meant by each Instruction. The changes in the password are shown visually in red color.

Step #	**Instruction**	**Example #1:** Firstname: *angelito* Lastname: *mercado*	**Example #2:** Firstname: *ron* Lastname: *sy*	**Example #3:** Firstname: *ik* Lastname: *tar*
1.	Initialize the password as a concatenation of Firstname, underscore and Lastname.	angelito_mercado	ron_sy	ik_tar
2.	If the length of the password is less than 8, add new characters at the end of the string such that the length will become 8. The new characters to be added are chosen such that they correspond to the sequence of letters (in the English alphabet). The sequence starts with the next-letter-after-the-last-character in the Lastname. If the last character is 'z', the letters to be added cycle back to 'a'. Examples are shown in the last two columns of this table. **Be very careful in this step. The original null byte will be overwritten. Do NOT forget to add a null byte in the password! Your codes for the next steps will have a logical error when the null byte is missing.**	angelito_mercado	ron_syza	ik_tarst
3.	Capitalize the first character of the first name and last name.	Angelito_Mercado	Ron_Syza	Ik_Tarst
4.	Replace lower case vowels as follows: 'a' with '@', 'e' with '3', 'i' with '1', 'o' with '0' and 'u' with '^'. Upper case vowels should not be replaced.	Ang3l1t0_M3rc@d0	R0n_Syz@	Ik_T@rst
5.	Replace lower case consonants as follows: 's' with '*', and 't' with '?'. Upper case consonants should not be replaced.	Ang3l1?0_M3rc@d0	R0n_Syz@	Ik_T@r*?

SPECIFIC REQUIREMENTS:

- You have to complete the skeleton code PASSWORD-LASTNAME.c.
- **You should define SIX functions** - one of them is the for main(), and the other five functions correspond to each step of the algorithm.

SAMPLE RUNS:

Three example program interactions are shown below. The first line shows the input Firstname, the second line shows the input Lastname, and the succeeding lines are the password values after each step. Your program should produce exactly the same results. The sample outputs are encoded in three accompanying text files EXAMPLE1.TXT, EXAMPLE2.TXT and EXAMPLE3.TXT.

Sample Run #1:

```
angelito
mercado
angelito_mercado
angelito_mercado
Angelito_Mercado
Ang3l1t0_M3rc@d0
Ang3l1?0_M3rc@d0
```

Sample Run #2:

```
ron
sy
ron_sy
ron_syza
Ron_Syza
R0n_Syz@
R0n_Syz@
```

Sample Run #3

```
ik
tar
ik_tar
ik_tarst
Ik_Tarst
Ik_T@rst
Ik_T@r*?
```

RUN YOUR PROGRAM WITH OUTPUT REDIRECTION: Run your exe file in the command line with input redirection. You can store the result of your program into a text file by output redirection, For example, if your last name is SANTOS, then you can run your executable program with output redirection as:

C:\CCPROG2> PASSWORD-SANTOS > SANTOS-RESULT3.txt

The output of the exe file will be stored in SANTOS-RESULT3.txt file. A correct program should produce the same set of values that are in EXAMPLE3.txt using the "ik" and "tar" as input values respectively.

TESTING & SCORING:

- Your program will be black box tested by your instructor with different input values.
- A program with a compilation error will AUTOMATICALLY result into a score of 0. Make sure that there are no syntax/compilation errors in your solution.
- 5 points for correct implementation of main(), 5 points for correct implementation of the function for Step 1, and 10 points each for the correct implementation of functions for Steps 2 to 5. The score for an incorrect function implementation is 0. For example, a score of 40/50 will be earned if all functions except that for Step 5 are correct.

DELIVERABLES:

Submit/upload two files via Canvas before the indicated deadline:

1. your PASSWORD-LASTNAME.c source file
2. your LASTNAME-RESULT3.txt output text file (this is the output file of your program corresponding to Example #3 as described above).

Don't forget to change the filename to your own last name.

-- The End --