**2016320130**

**Hojeong Lee**

**Theory of Computation**

**RE Programming Report**

**1. Environment**

I used **‘Python’** code and **‘import re library’** to create a RE program that solves the problems.

**2. Problem 1**

**1) SNS link**

Problem 1 is to make a SNS link classifier from Hacker’s message.

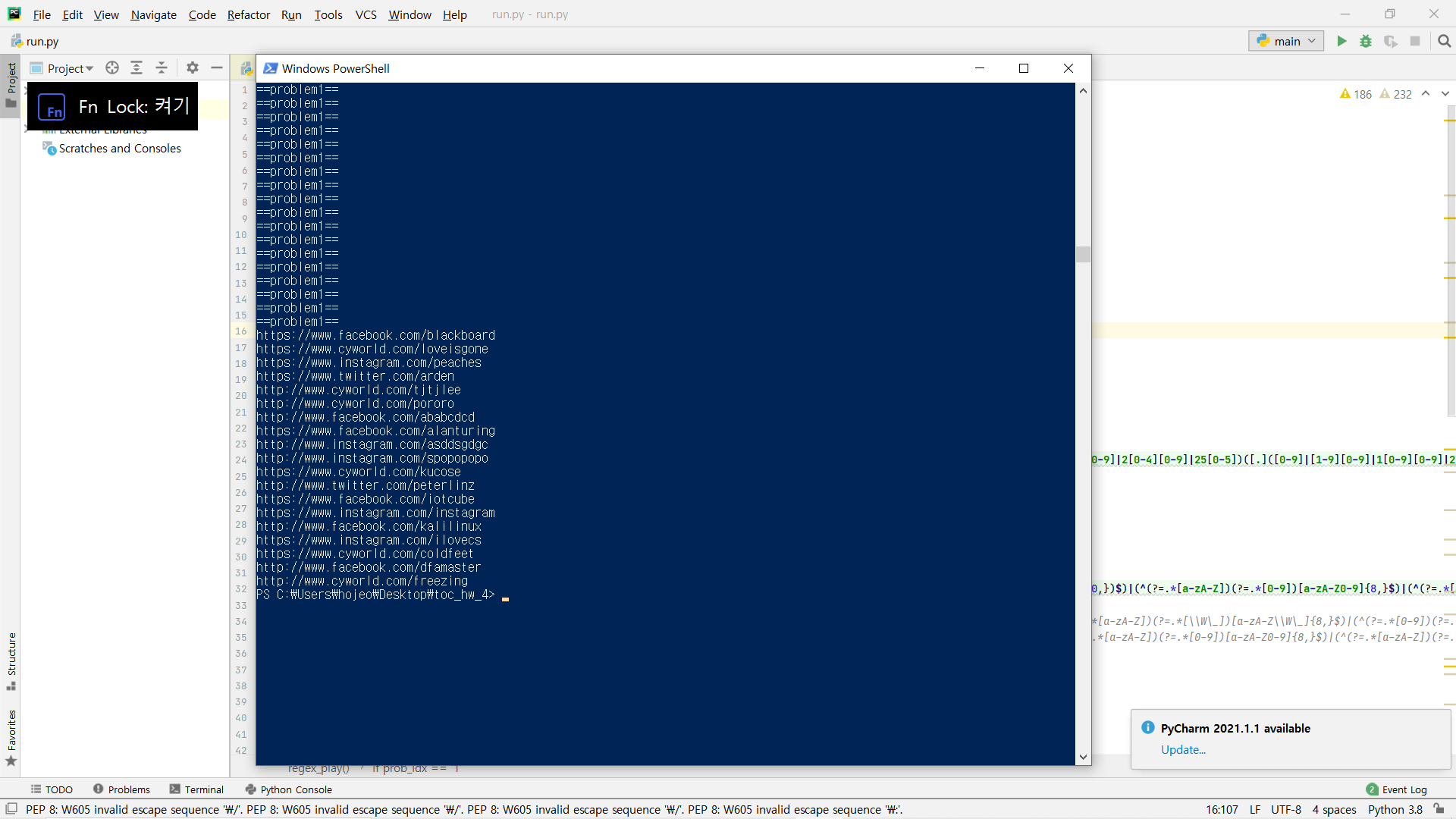
텍스트이(가) 표시된 사진

자동 생성된 설명

I wrote my code like above. The given condition said that ‘both uppercase and lowercase are allowed’ so I compile my code with ‘re.IGNORECASE’. It makes me easier to think about the conditions.

Regex starts with ‘^http[s]?’ which means that check if the input string starts with http or https. Then, the next part of the form of SNS link is like ‘://www.’ so regex continues with ‘\:\/\/www[.]’ which has the meaning of check if the string continues with ‘://www.’ In addition, the types of SNS links are cyworld, facebook, instagram, twitter so the regex ‘((facebook)|(instagram)|(cyworld)|(twitter))’ part checks the string satisfied the conditions. Following regex ‘[.]com\/’ part checks string ‘.com/’ part. Finally, last part of the SNS link is user ID which can be any length of alphabets so I assumed that length 0 can be accepted. The regex regarding this is ‘[a-zA-Z]\*$’ which means that any length of strings consist of alphabets can be accepted.

**2) Results**



The results showed that 6 cyworld, 6 facebook, 5 instagram, 2 twitter links. So the password to go to the problem 2 was 6652!

**3. Problem 2**

Problem 2 is to find e-mail address, IP address, and credit card number. I wrote the code like above, and I will explain about it one by one.

텍스트이(가) 표시된 사진

자동 생성된 설명

**1) E-mail address**

텍스트이(가) 표시된 사진

자동 생성된 설명

First part of the regex consider about the e-mail address. The condition said that e-mail address’s first letter must be an alphabet so the regex starts with ‘^[a-zA-Z]’ The rest part of string before @ can be any length of number, alphabet, .(dot) and \_(underscore) so following regex is ‘[a-zA-Z0-9\\_\.]\*’ which means same as the condition. After @ part, the form can be any length (more than 0) of lowercase alphabets so ‘[a-z]+’ part checks the condition. Finally, the last part of the address should be ‘.ac.kr’ or ‘.com’ or ‘.net’ or ‘.co.kr’ so the regex ‘(([.]ac[.]kr)|([.]com)|([.]net)|([.]co[.]kr))$’ part checks about it.

**2) IP address**

텍스트이(가) 표시된 사진

자동 생성된 설명

Second part of the regex consider about the IP address. The form should be ‘A.B.C.D’ and A, B, C, D should be a number in 0 ~ 255. First, I thought it will be much easier if I separated the string into several parts. So I separate ‘A’ and ‘.B’, ‘.C’ and ‘.D’

‘A’ should be a number in 0~255, so the regex ‘^([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])’ part consider the 0~9, 10~99, 100~199, 200~249, 250~255 to satisfy the condition. Now, when we think about the rest parts like ‘.B’, ‘.C’ and ‘.D’, the B, C, and D should be a number in 0~255 so the expression for those things are same. When we add ‘.’ in front of the number, it could satisfy the condition. In addition, we can make ‘.B.C.D’ part just repeat same expression three times so the regex of this part is ‘([.]([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])){3}’ This expression finally satisfy the condition of IP address.

**3) Credit card number**

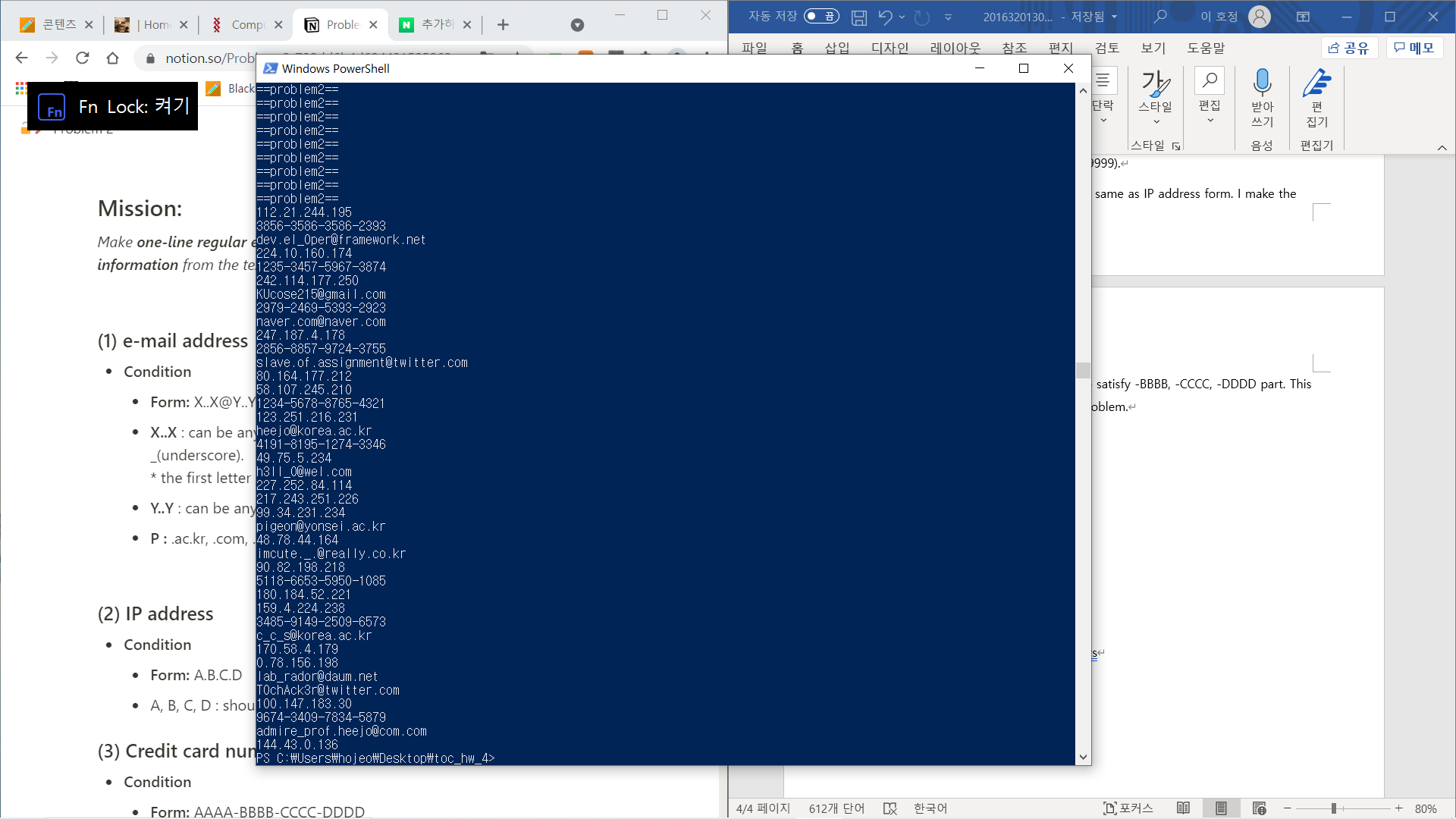
텍스트이(가) 표시된 사진

자동 생성된 설명

The last part of the regex considers about the credit card number. The form of the credit card number is ‘AAAA-BBBB-CCCC-DDDD’ and AAAA, BBBB, CCCC, DDDD should be a 4-digit number(0000~9999).

I think credit card number form is almost same as IP address form. I make the regex for AAAA, then just repeat 3 time to satisfy -BBBB, -CCCC, -DDDD part. This approach makes me easier to solve this problem.

**4) Results**



The results are consisted of the exact form of e-mail addresses, IP addresses, and credit card numbers.

The password to go to the problem 3 is the last 4 letters of third valid credit card number which is ‘2923’ from the string ‘2979-2469-5393-2923’

**4. Problem 3**

Now, it is the final problem! In this problem, I make one-line regular expression that checks the password which follows the password rules.

I wrote my regex in one-line, but it was too long to capture. So, I will explain about my regular expression one by one.

**1) Check for 3 or more consecutive characters**

텍스트이(가) 표시된 사진

자동 생성된 설명

First, I check the 3 or more consecutive characters condition. This can be done with the regular expression above. This rejects the string which has 3 consecutive characters, so it will also automatically reject the string which has more than 3 consecutive characters.

**2) Check for whitespaces**

텍스트이(가) 표시된 사진

자동 생성된 설명

I check if the password has whitespaces or not because the conditions said that the type of characters can be alphabets, digits, and special characters which does not include whitespaces.

**3) 10 or more length passwords**

텍스트이(가) 표시된 사진

자동 생성된 설명

In case of 10 or more length passwords, only one type of character can be used except digits. It means that 10 or more alphabets can be accepted, and 10 or more special characters can be accepted. So, I made the above regex which regarding those conditions. First half of the regex consider the special characters only cases. [\W] means that characters which are none of alphabets, digits, and \_(underscore). So [\W\\_] contains all characters except alphabets and digits. Remaining part of the regex consider the alphabets only cases.

**4) 8 or more (including 10 or more) length**

In those cases, the password should be consisted of 2 or more types of characters. In other words, password should be in 4 cases which are (1) alphabets and digits (2) alphabets and special characters (3) digits and special characters (4) alphabets, digits, and special characters.

**(1) Check if string has alphabets or not, and digits or not**

텍스트이(가) 표시된 사진

자동 생성된 설명

**(2) Check if string has alphabets or not, and special characters or not**

텍스트이(가) 표시된 사진

자동 생성된 설명

**(3) Check if string has digits or not, and special characters or not**

텍스트이(가) 표시된 사진

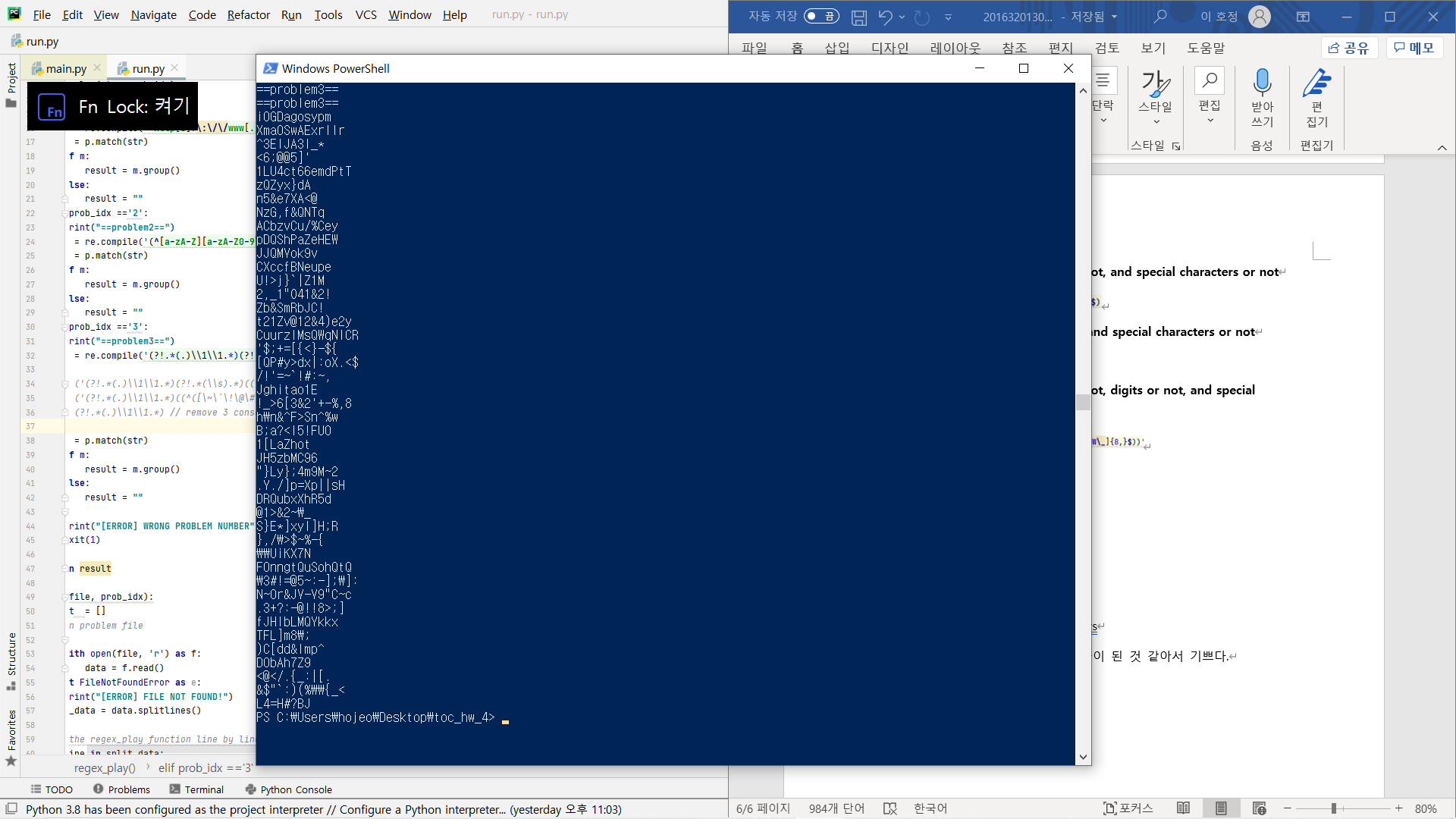
자동 생성된 설명

**(4) Check if string has alphabets or not, digits or not, and special characters or not**

텍스트이(가) 표시된 사진

자동 생성된 설명

**5) Results**



The results are consisted of the passwords which satisfy conditions. For example, iOGDagosypm and XmaOSwAExrllr are the passwords which satisfy the rule number **3)**. Also, <6;@@5]' and zQZyx}dA are the passwords which satisfy the rule number **4)**.

**5. Ending the report**

First of all, I really enjoyed the homework. Actually, I always thought that I was not good at programming. But after I solved all problems of this course, I gain confidence about the programming skills. The homework problems were not so easy. There would be no situations like this if there weren’t interesting stories and TA who reply questions about homework so kindly. I am happy to improve my programming skills.