**KATHMANDU UNIVERSITY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



Lab Work 2

COMP 316

SUBMITTED BY: SUBMITTED TO:

Arun Regmi Sushil Nepal

Roll No: 66 DOCSE

Computer Science (3rd Year/ 1st Sem)

Date of Submission: February 12, 2020.

# 

**Problem**

1. Write a program to implement regular expressions  
   Program Feature

* Write Program for given regular expressions
  + [a-z | A-Z] [a-z|A-Z|0-9] \*
* Input valid and invalid string:
  + valid: aB1, AZ, Z9
  + invalid: 9B,9\_AB5
* Output:
  + 'String accepted' for valid string
  + 'String rejected' for invalid string.

1. Write a Program to validate valid string from the given set of regular expressions. Your program should import .txt file that contains at least 3 regular expressions written in separate lines.

**Key Features**

The program “lab1.py” compiles a given regex pattern given by the user in the program and “lab2.py” reads the regex pattern from the “reg.txt”. The compiled regex is then used to check whether the string provided from the user in valid for the given pattern or not.

**Code:**

* Lab1.py

import re

p = re.compile('[a-z|A-Z][A-Z|a-z|0-9]\*')

s1 = "aB1, AZ, Z9"

match = p.match(s1)

if match==None:

print("String s1 is not accepted.")

else:

print("String s1 is accepted.")

s2 = "9B, 9\_AB5"

match = p.match(s2)

if match==None:

print("String s2 is not accepted.")

else:

print("String s2 is accepted.")

## Lab2.py

import re

def regexCompile(pattern):

return re.compile(pattern)

c\_regex = regexCompile("[a-z|A-Z][a-z|A-Z|0-9]\*")

in\_string ="Welcome to the planet 338C Let's host @ hostages."

allRegex = [line.rstrip() for line in open("reg.txt")]

for rexp in allRegex:

c\_regex = regexCompile(rexp)

match = c\_regex.match(in\_string)

if match==None:

print ("String Is Not Valid")

else:

print ("String Is Valid")

reg.txt

[a-z|A-Z][a-z|A-Z|0-9]\*

[\w\.-]+@[\w\.-]+

([\d]{4}[-][\d]{2}[-][\d]{

**Description:**

The function regexCompile function returns a compile regex object.

allRegexExp stores all the regex pattern contained in the “reg.txt”. We then loop through each pattern and compile that pattern. After that we check whether the given string pattern in valid or invalid using that compiled pattern.

**Output:**

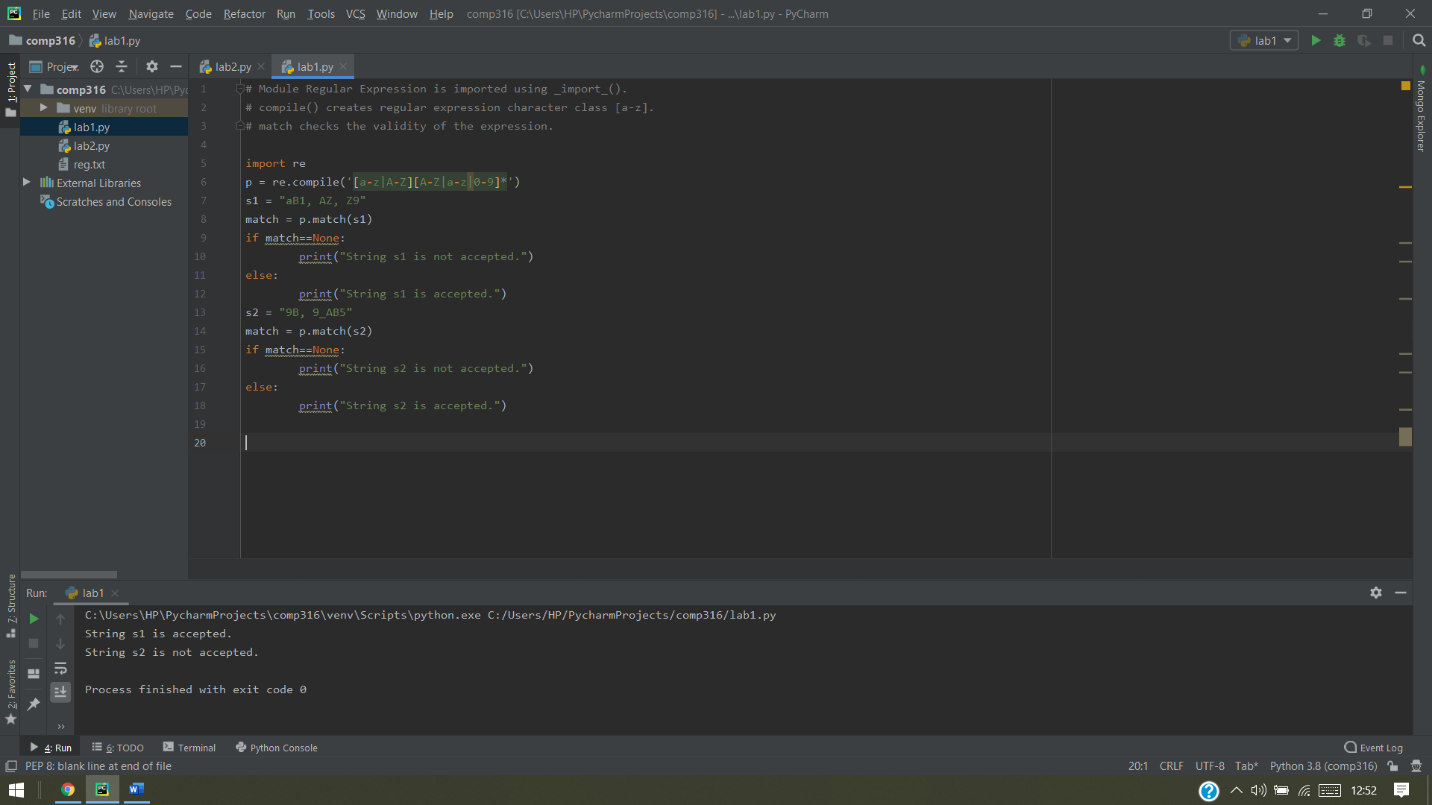


Image 1: Output for program 1

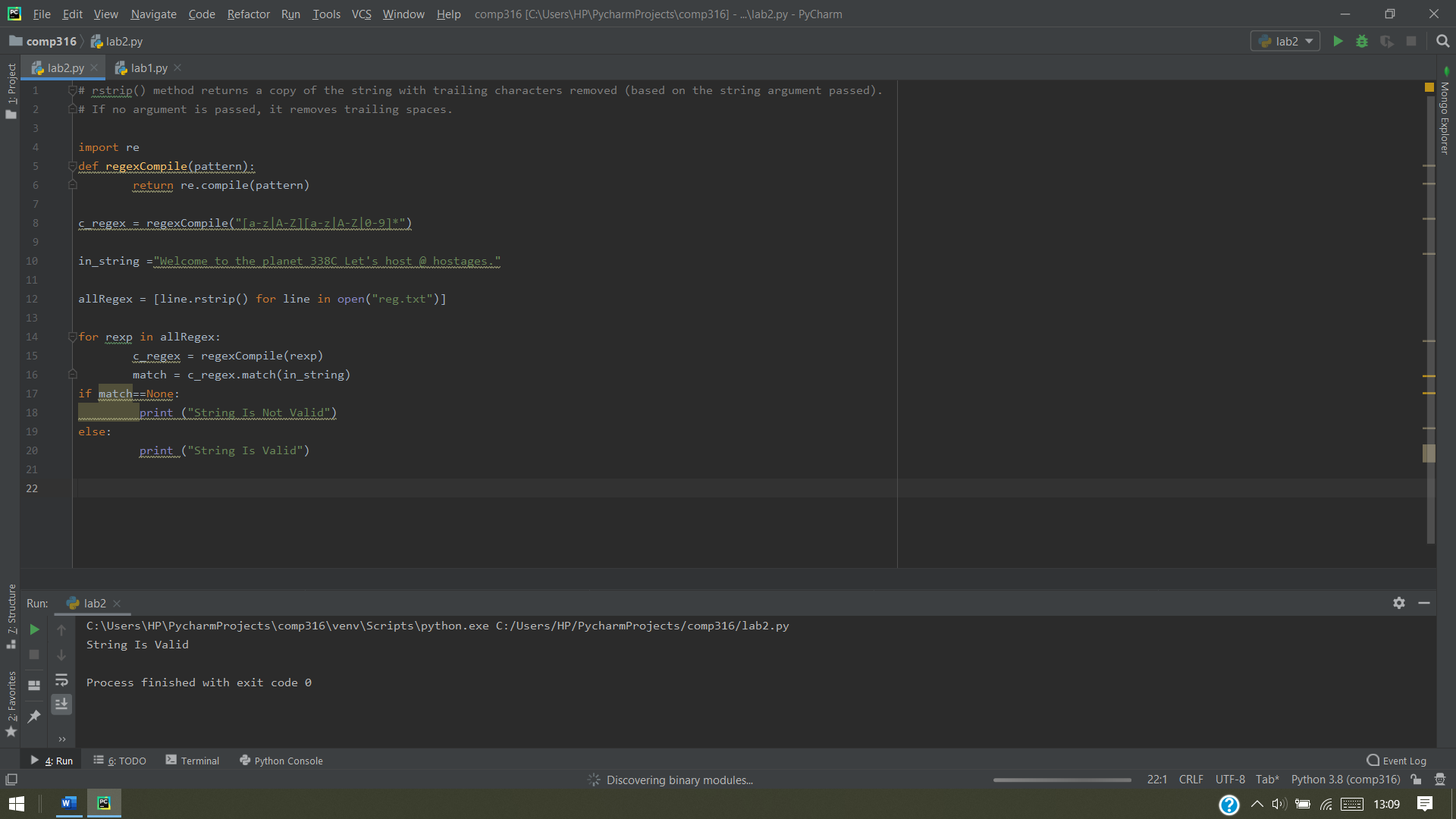


Image 2: Output for program 2