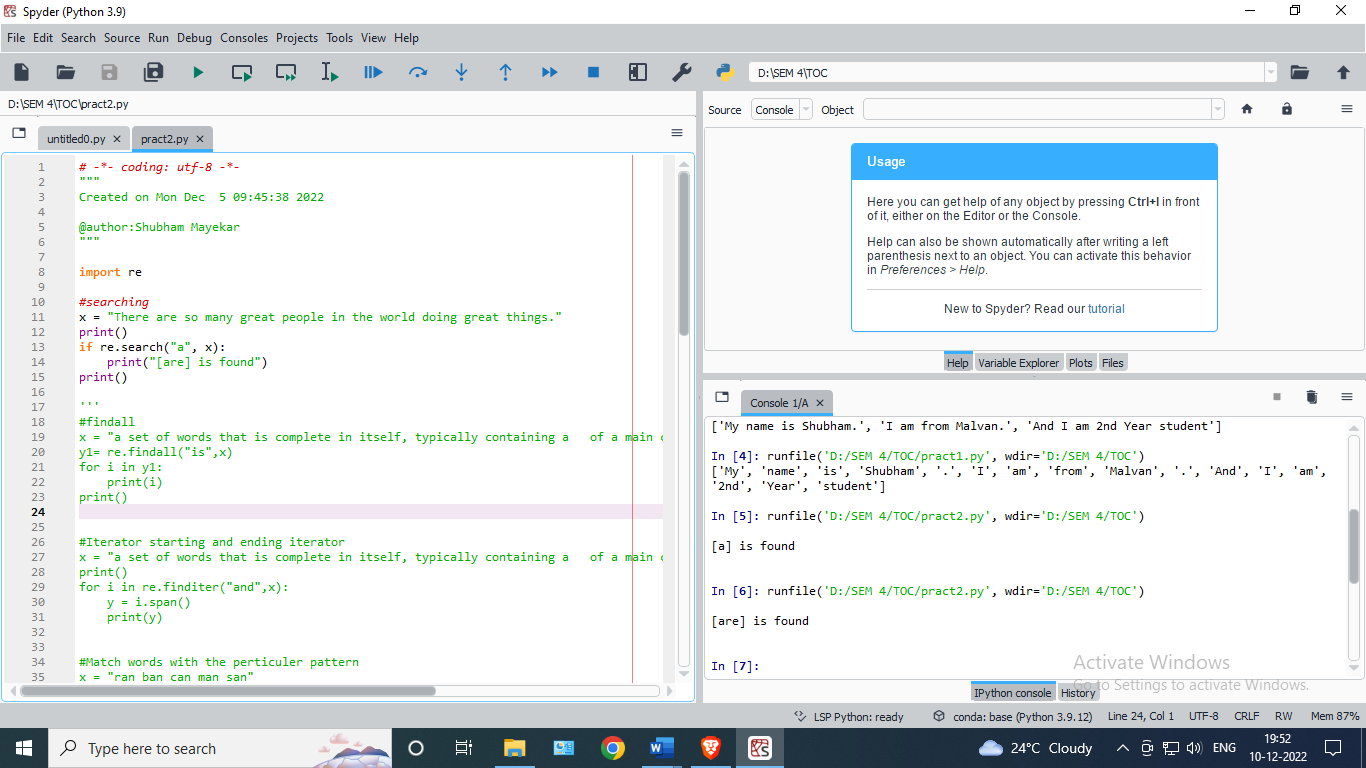
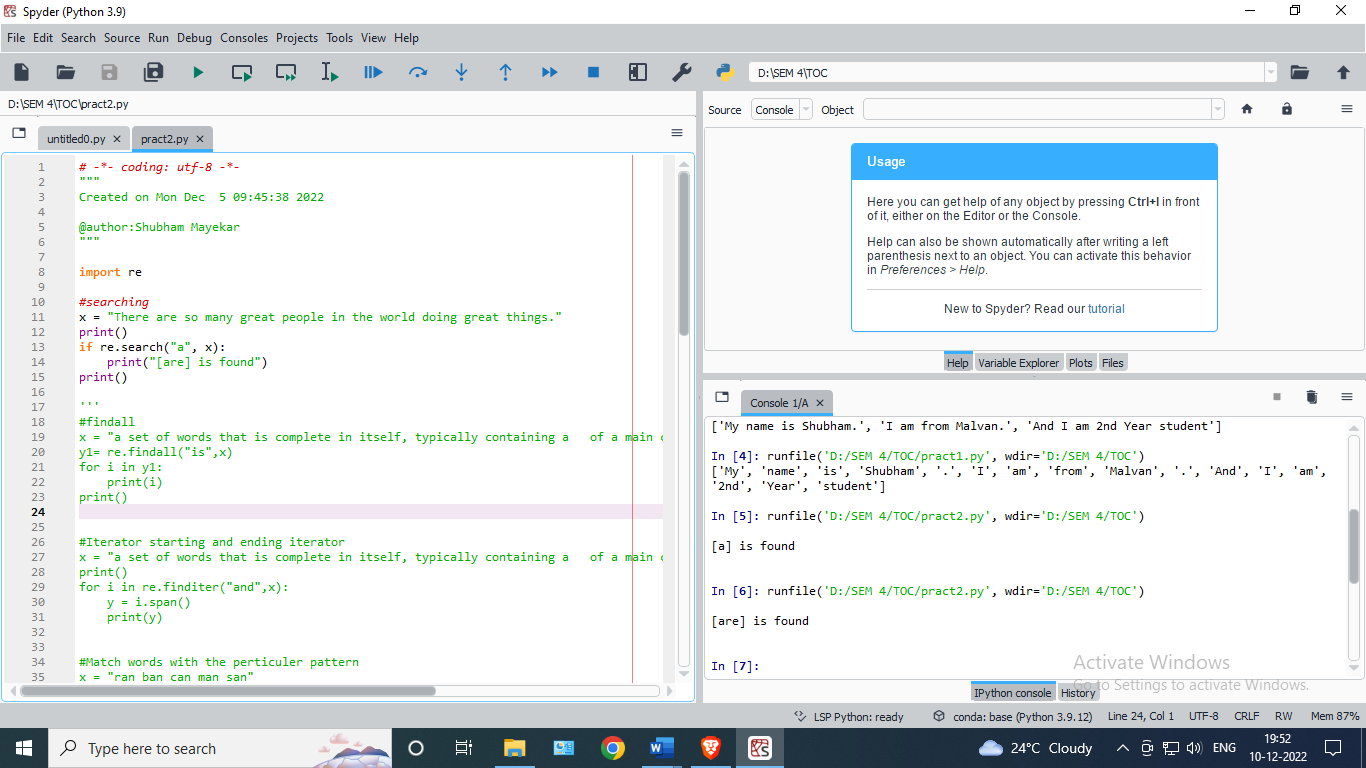
**Practical no 2**

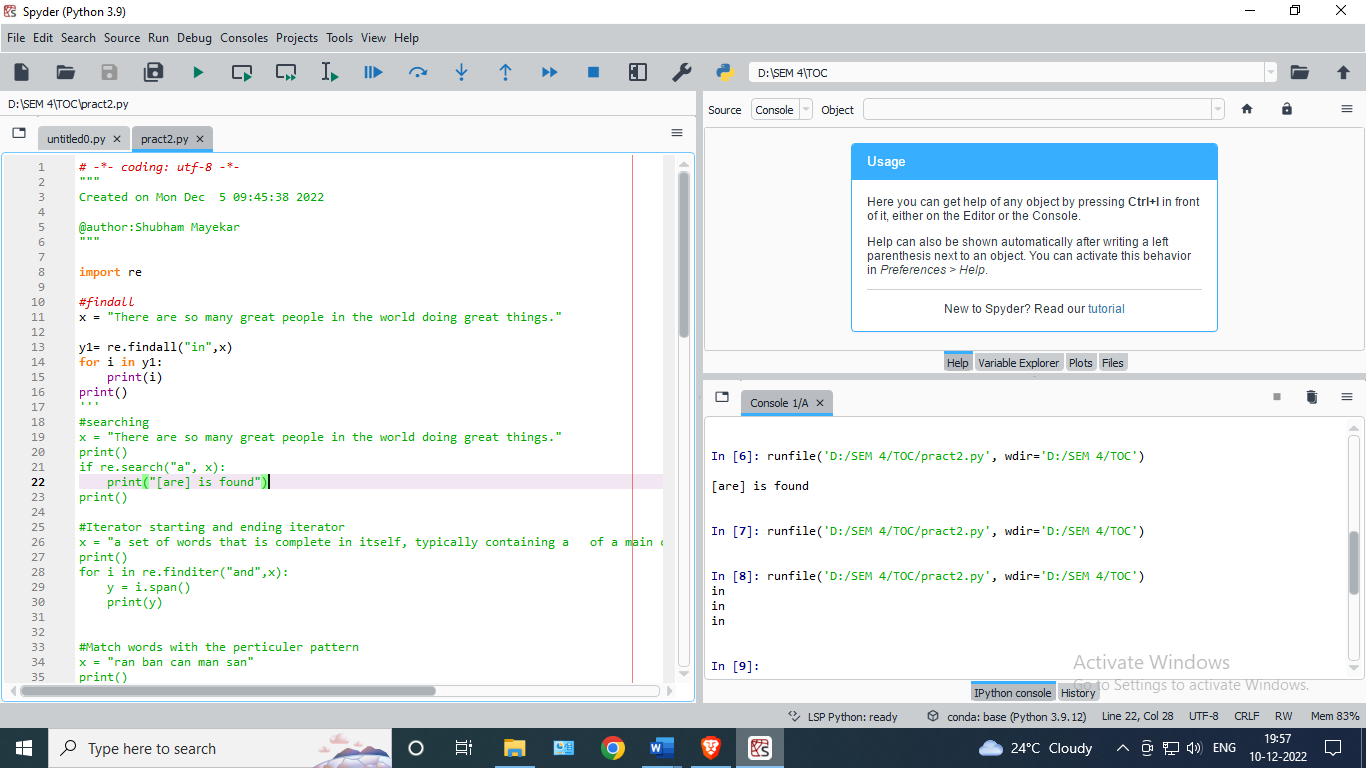
* Write a program for genrating regular expression for regular grammar.
* **Searching :**



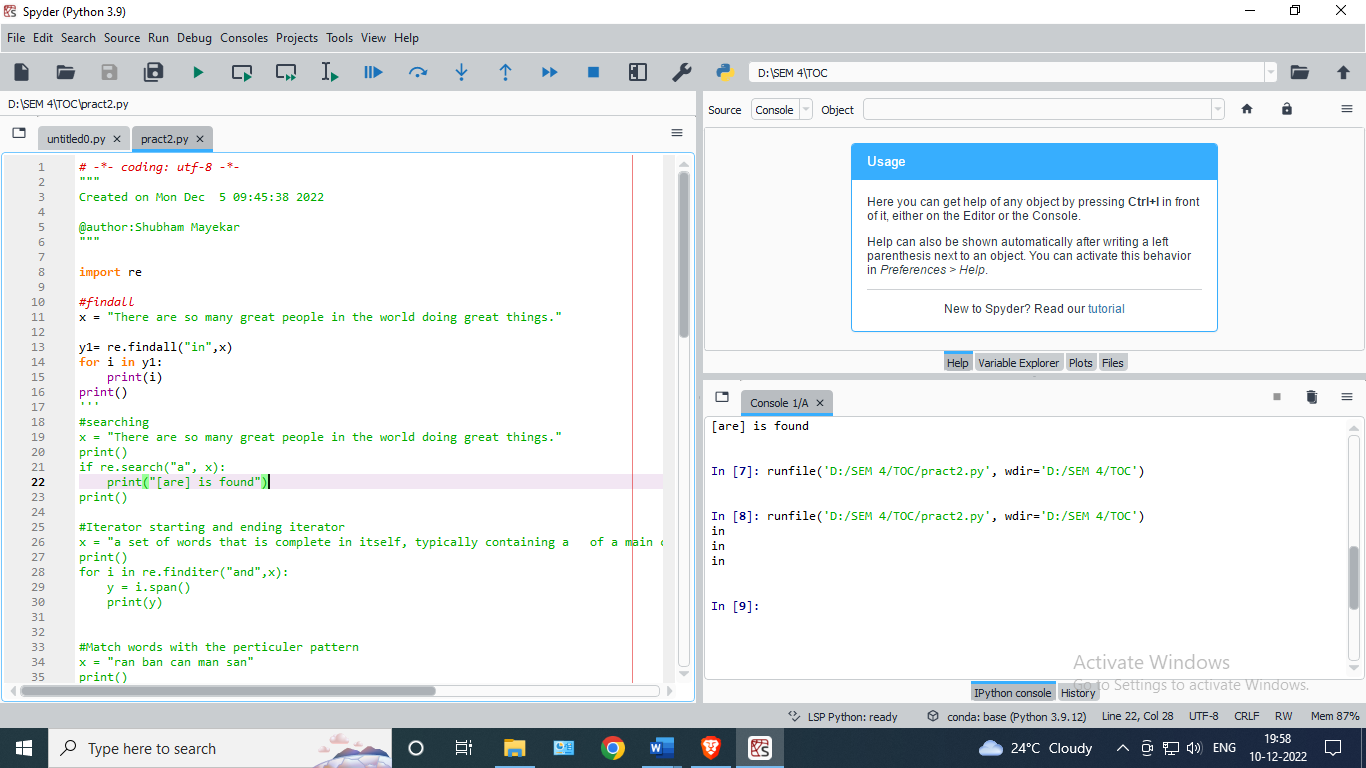
* **Output:**



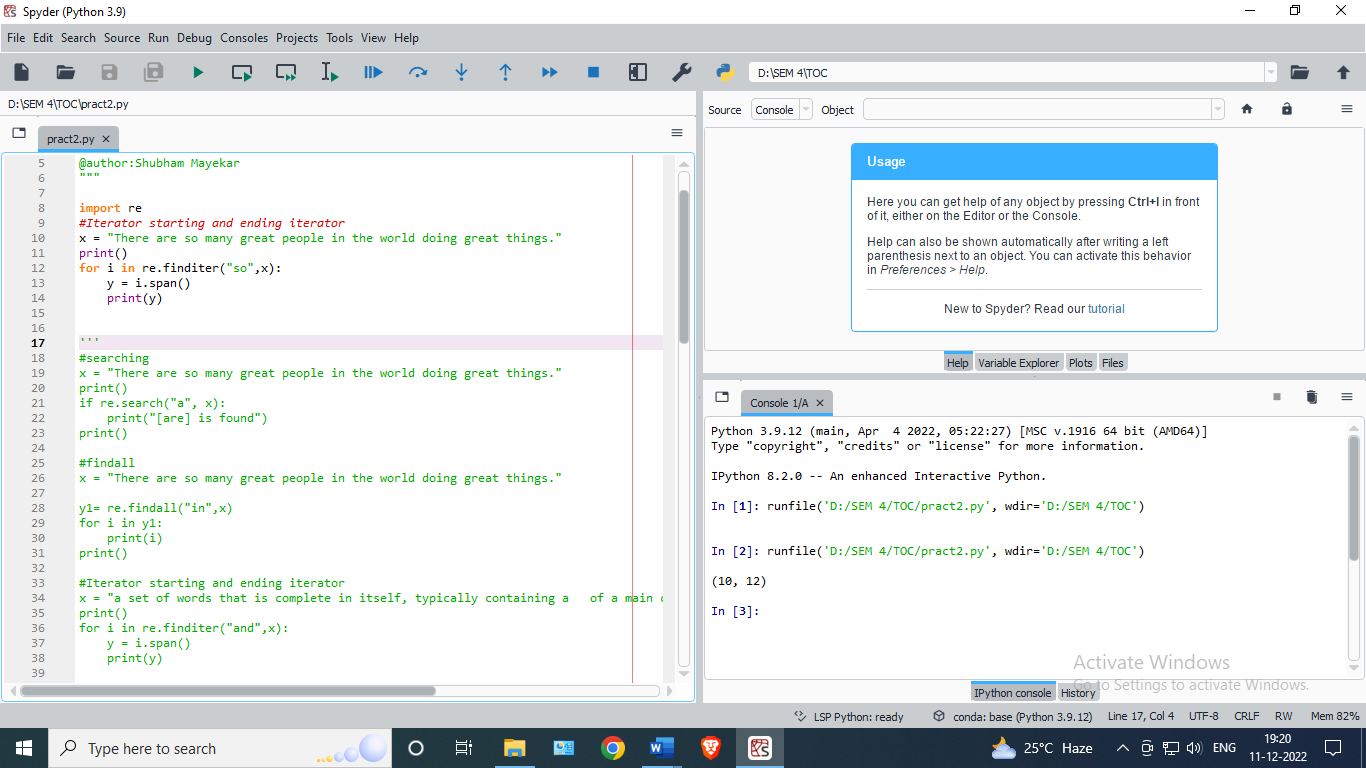
* **Find All :**



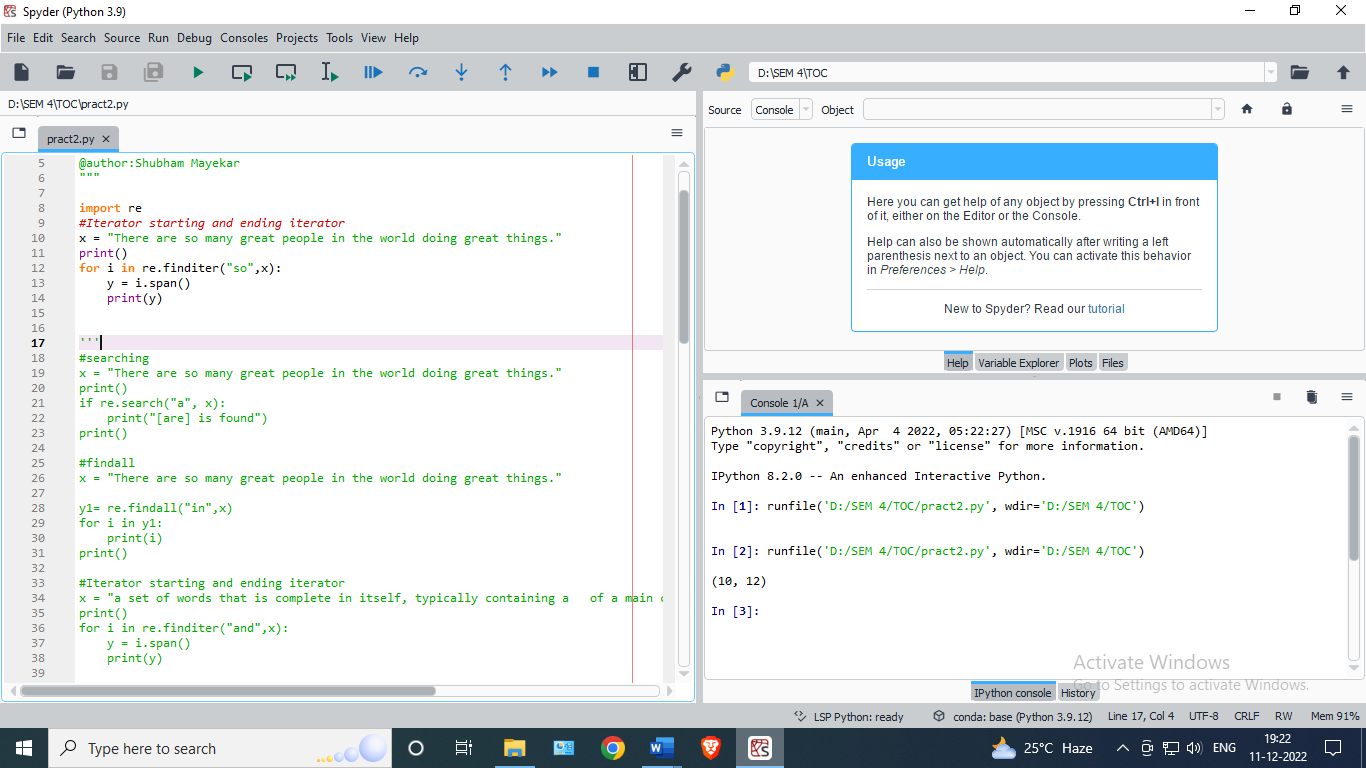
* **Output:**



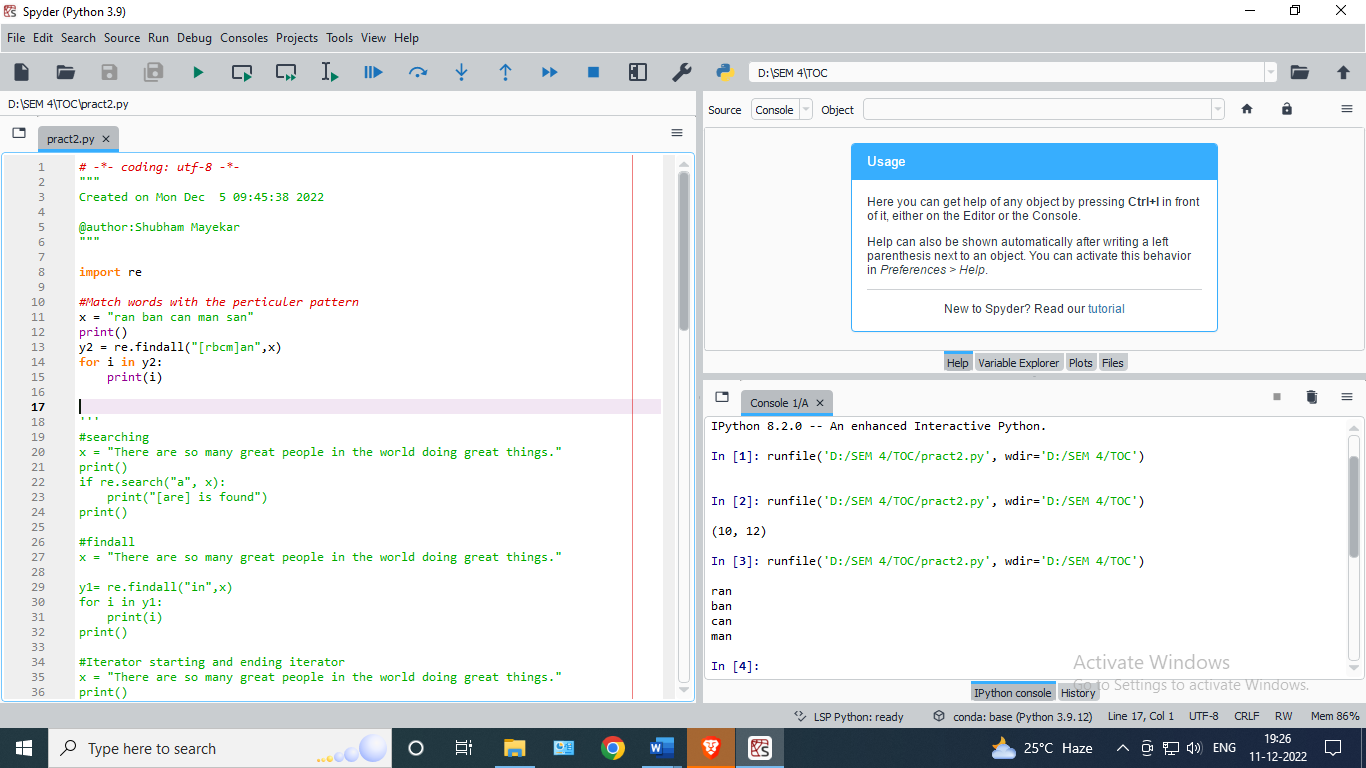
* **Iterator starting and edingiteartor :**



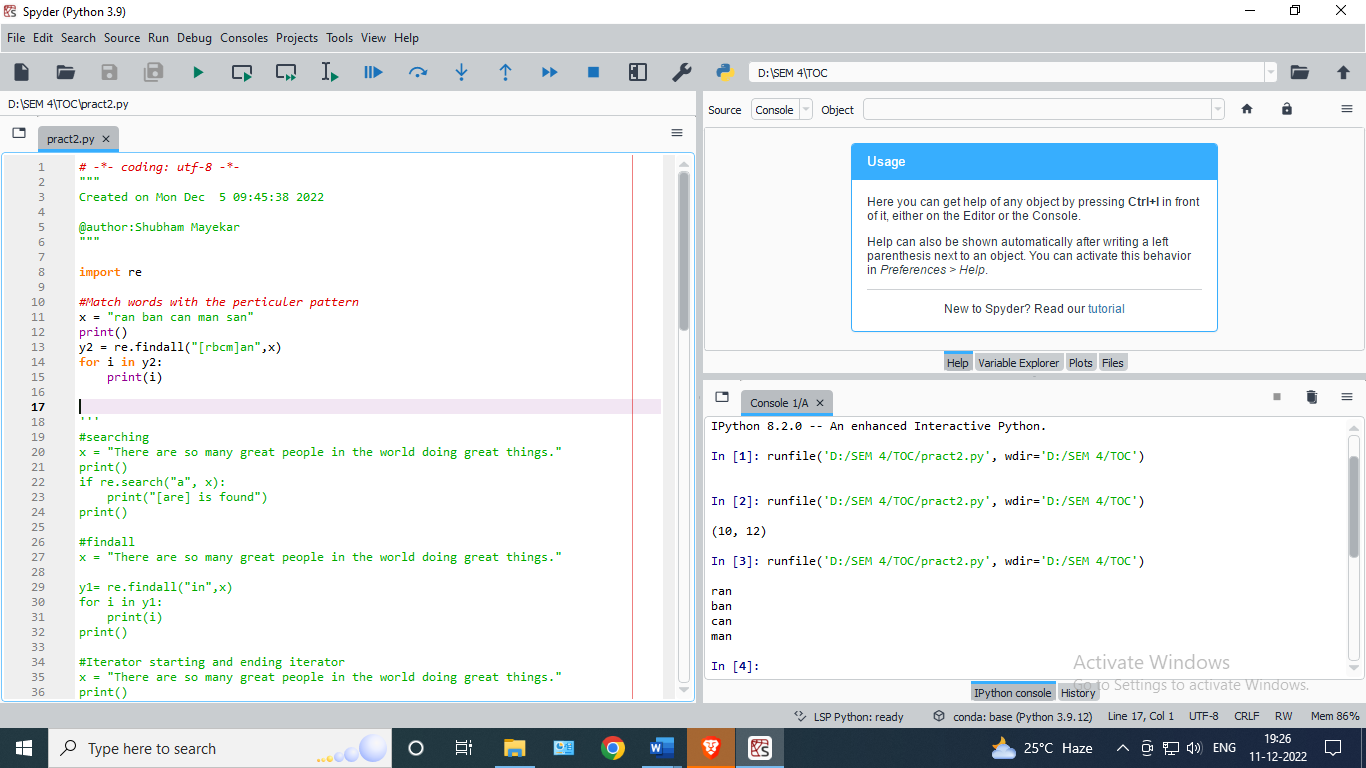
* **Output:**



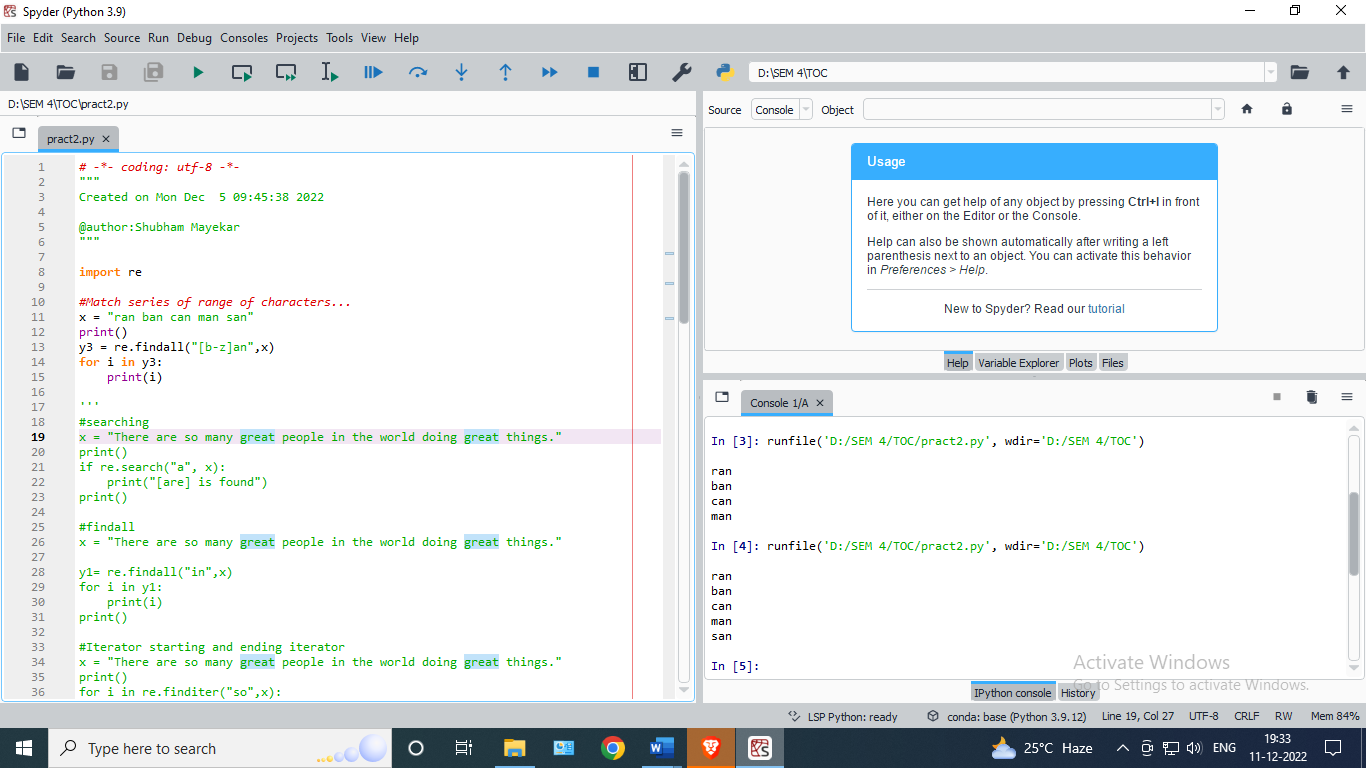
* **Match words with perticulerpattern :**



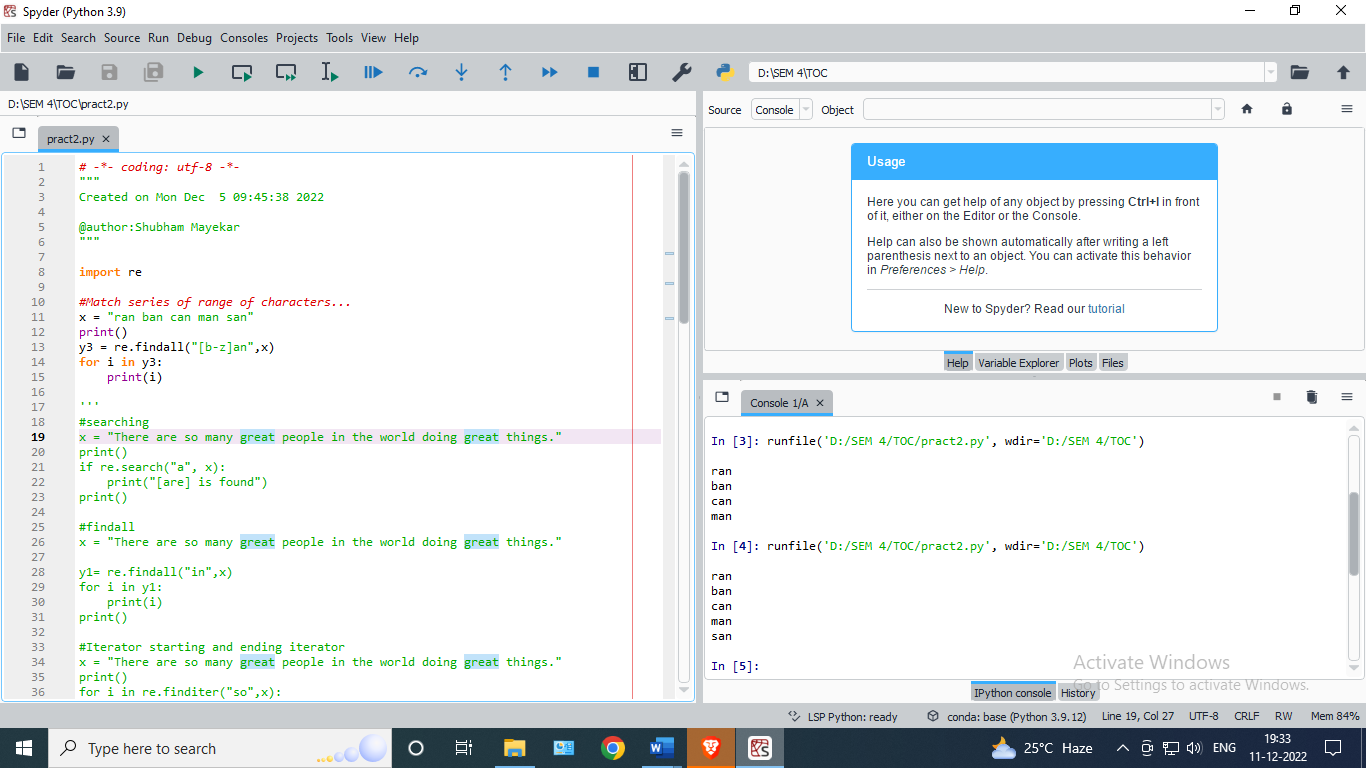
* **Output:**



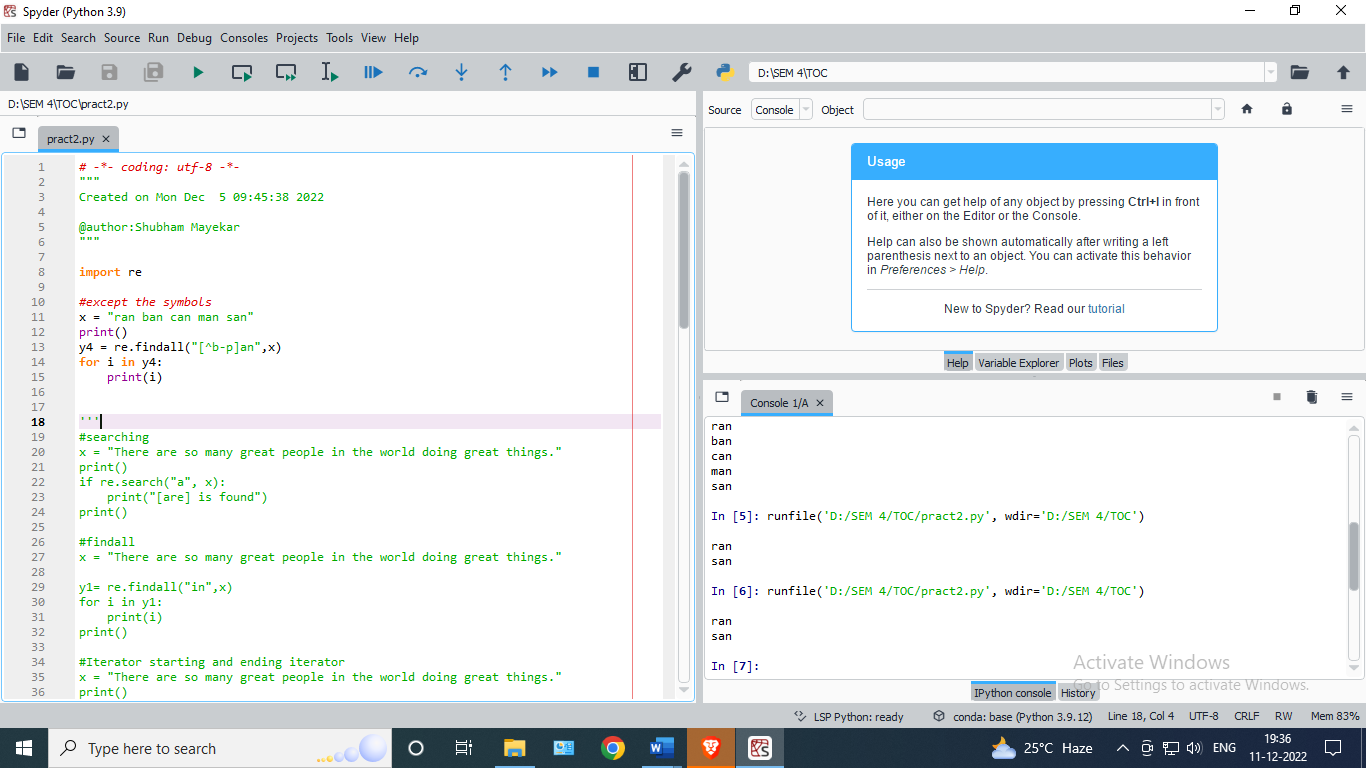
* **Match series of range of character :**



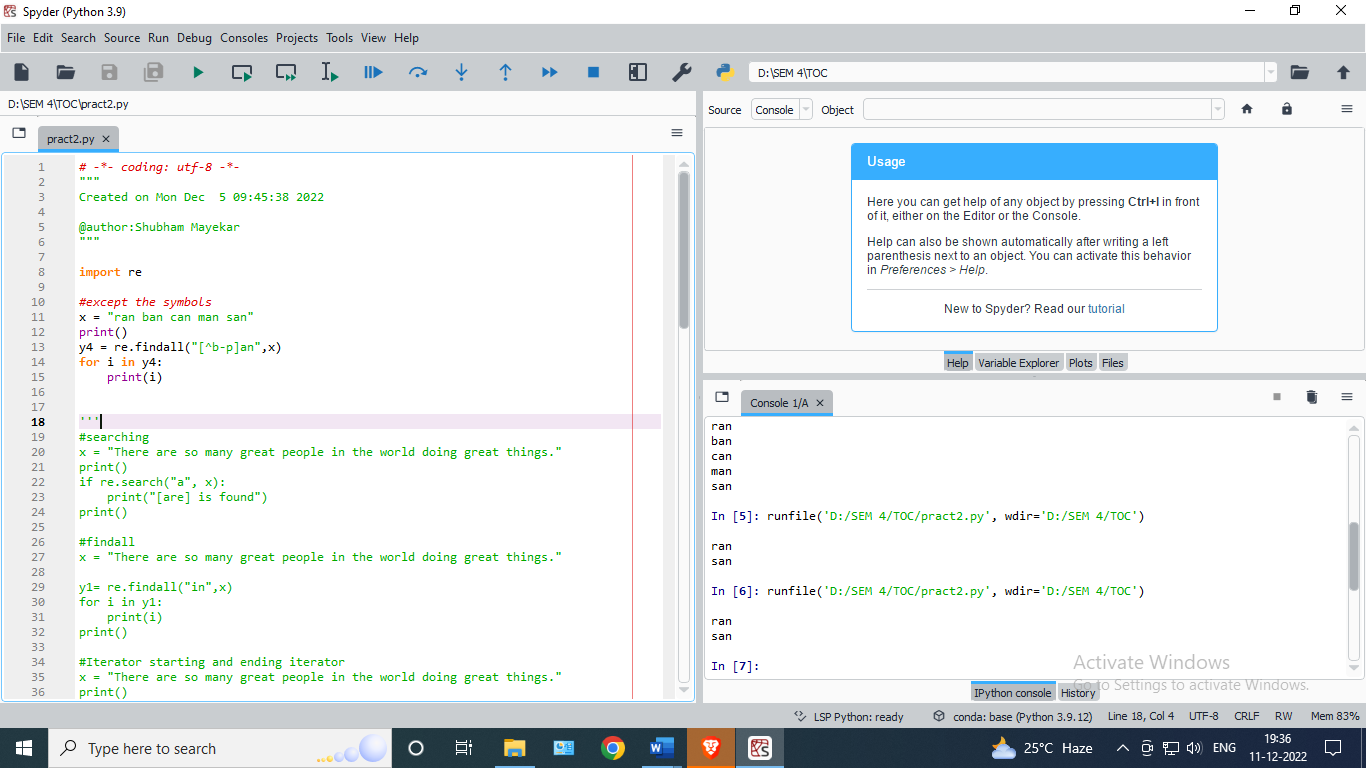
* **Output:**



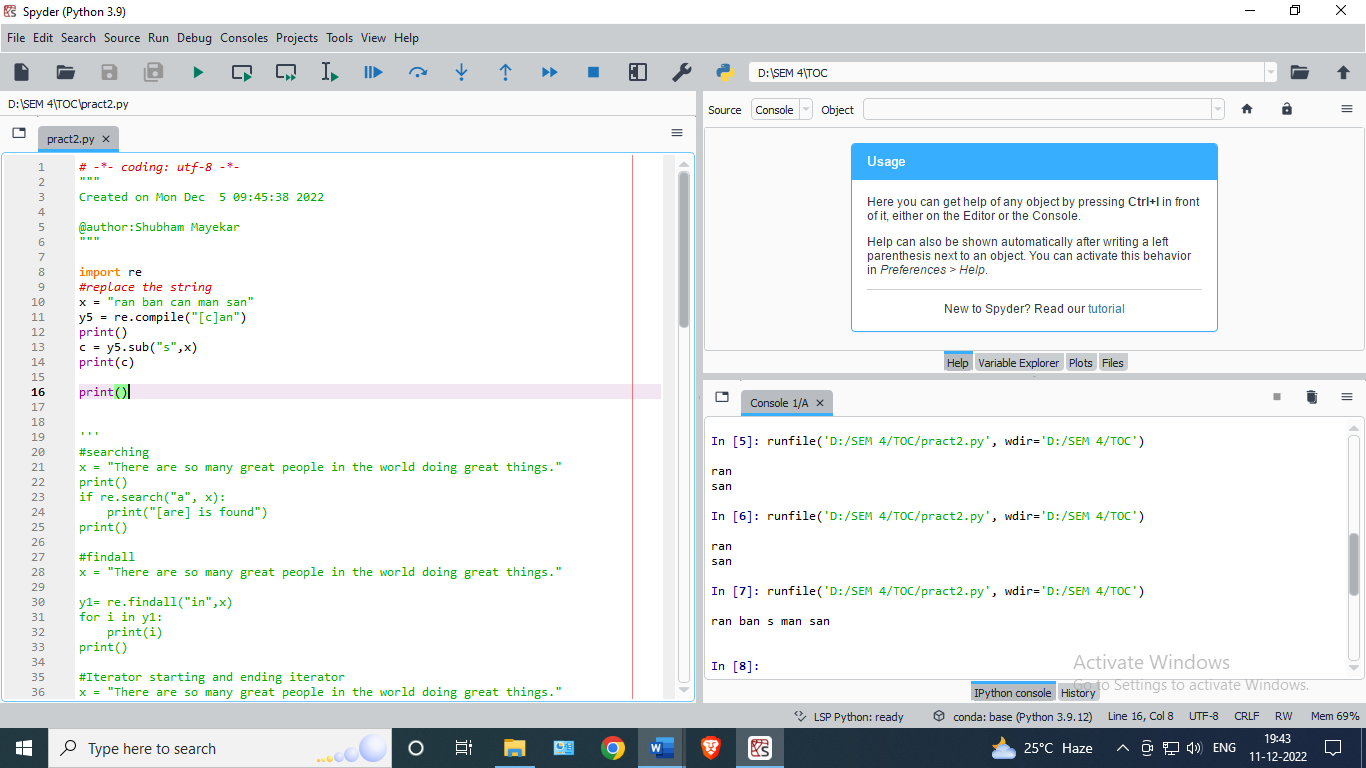
* **Except the symbol:**



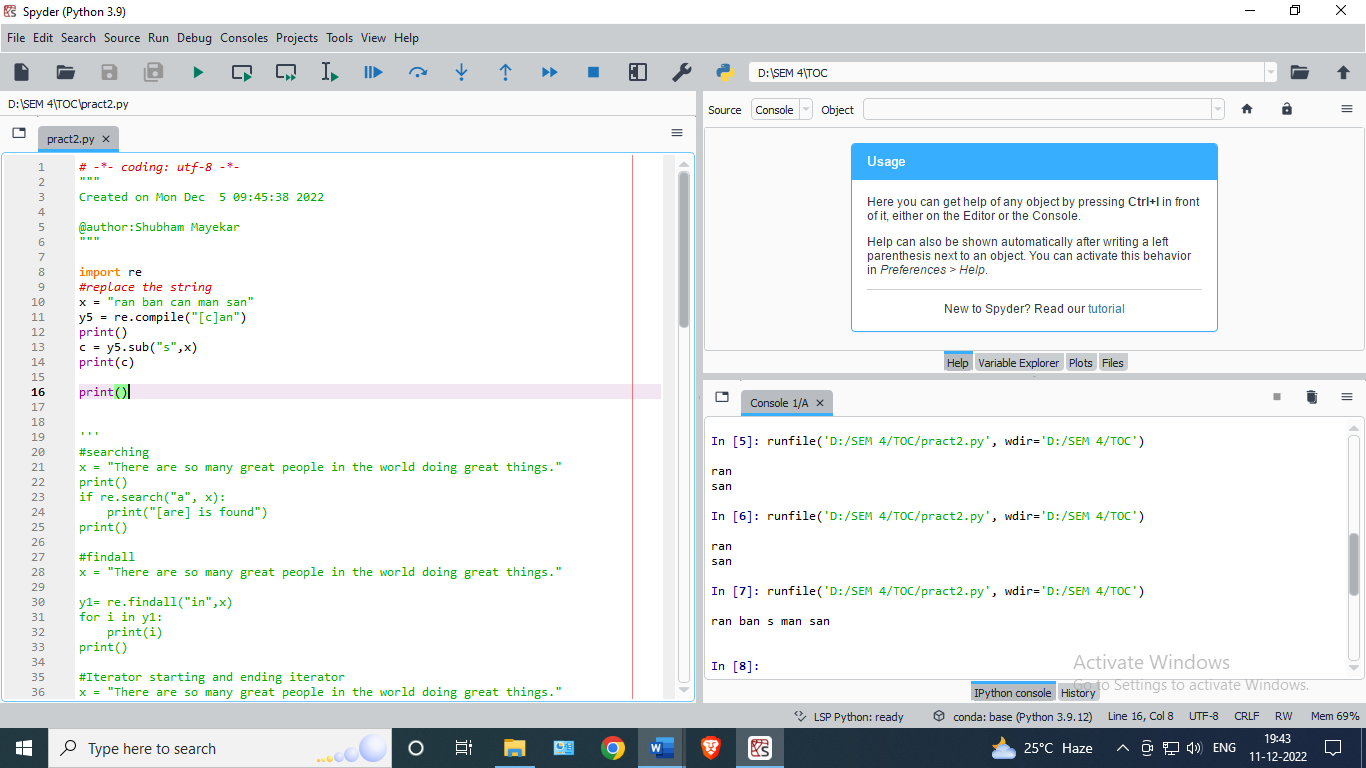
* **Output:**



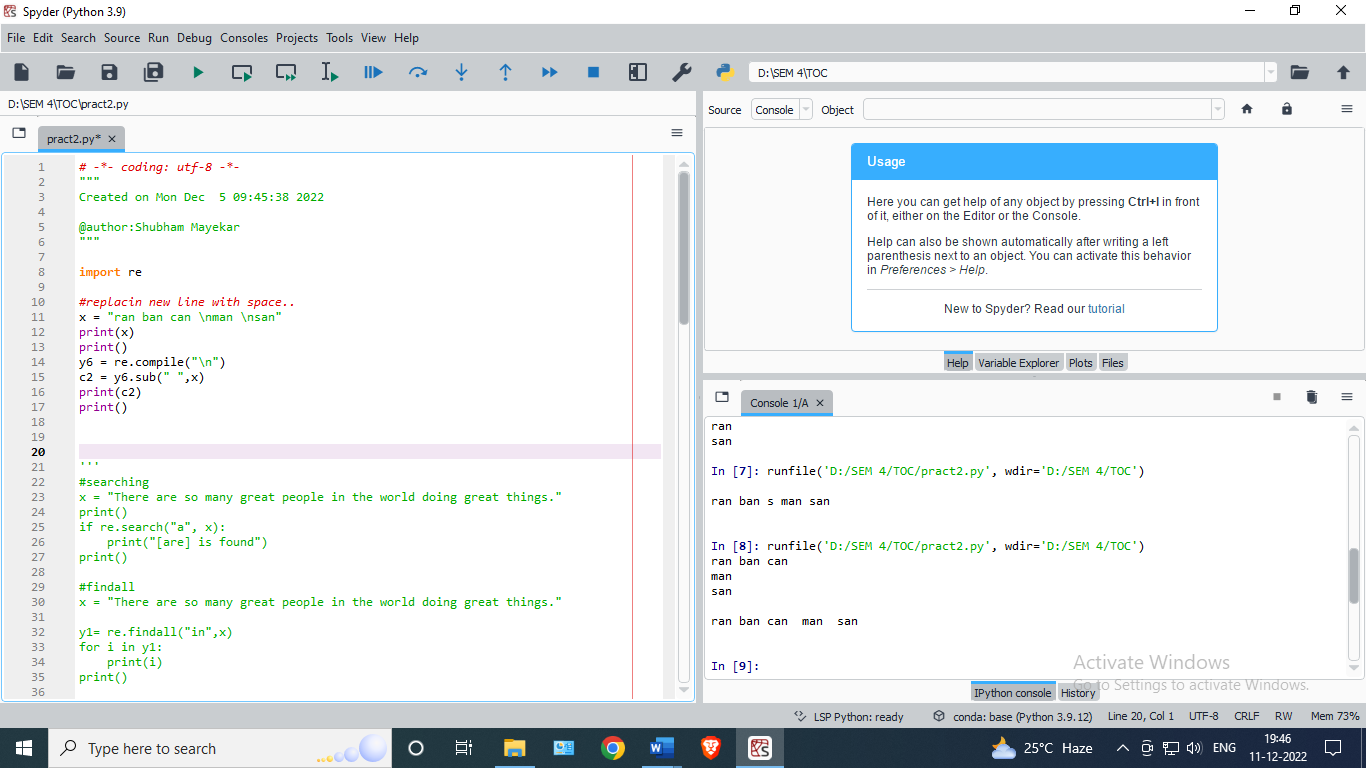
* **Replace the string :**



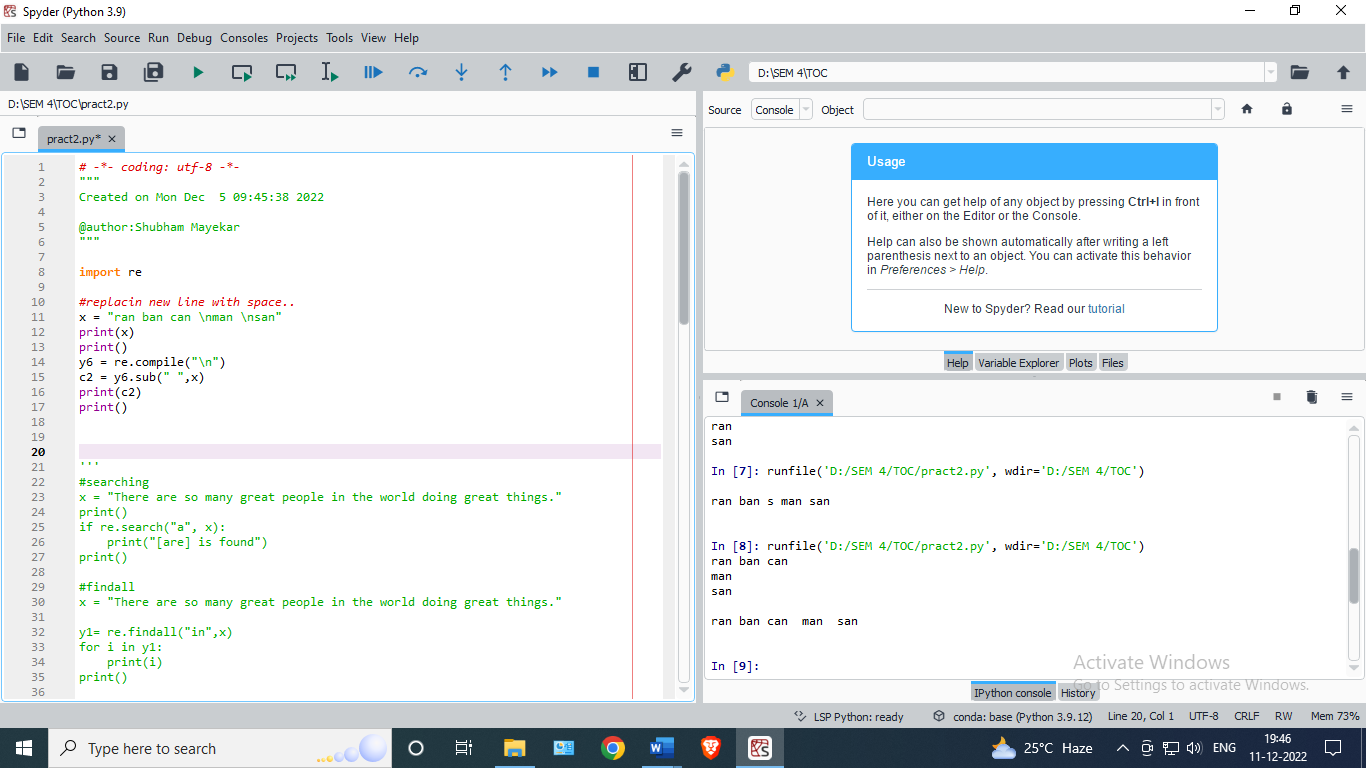
* **Output:**



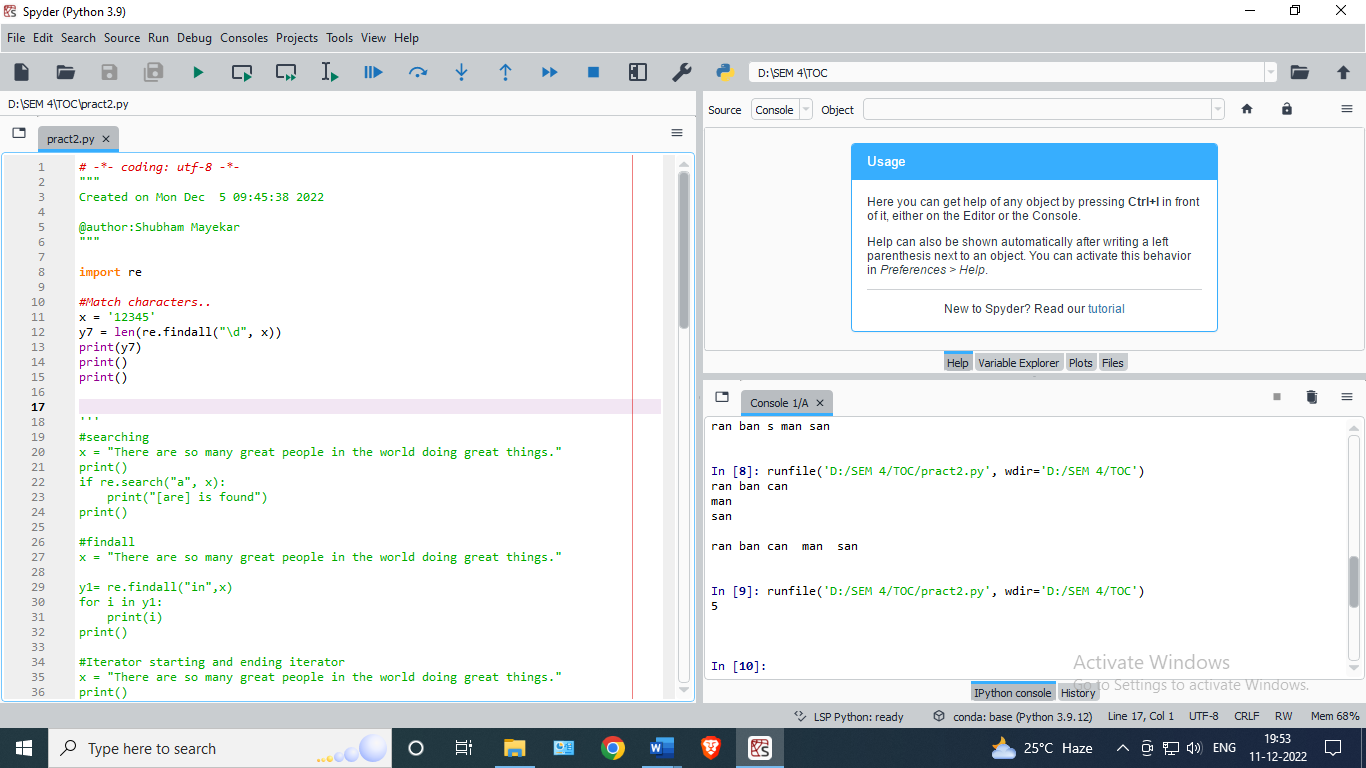
* **Replace new line with space :**



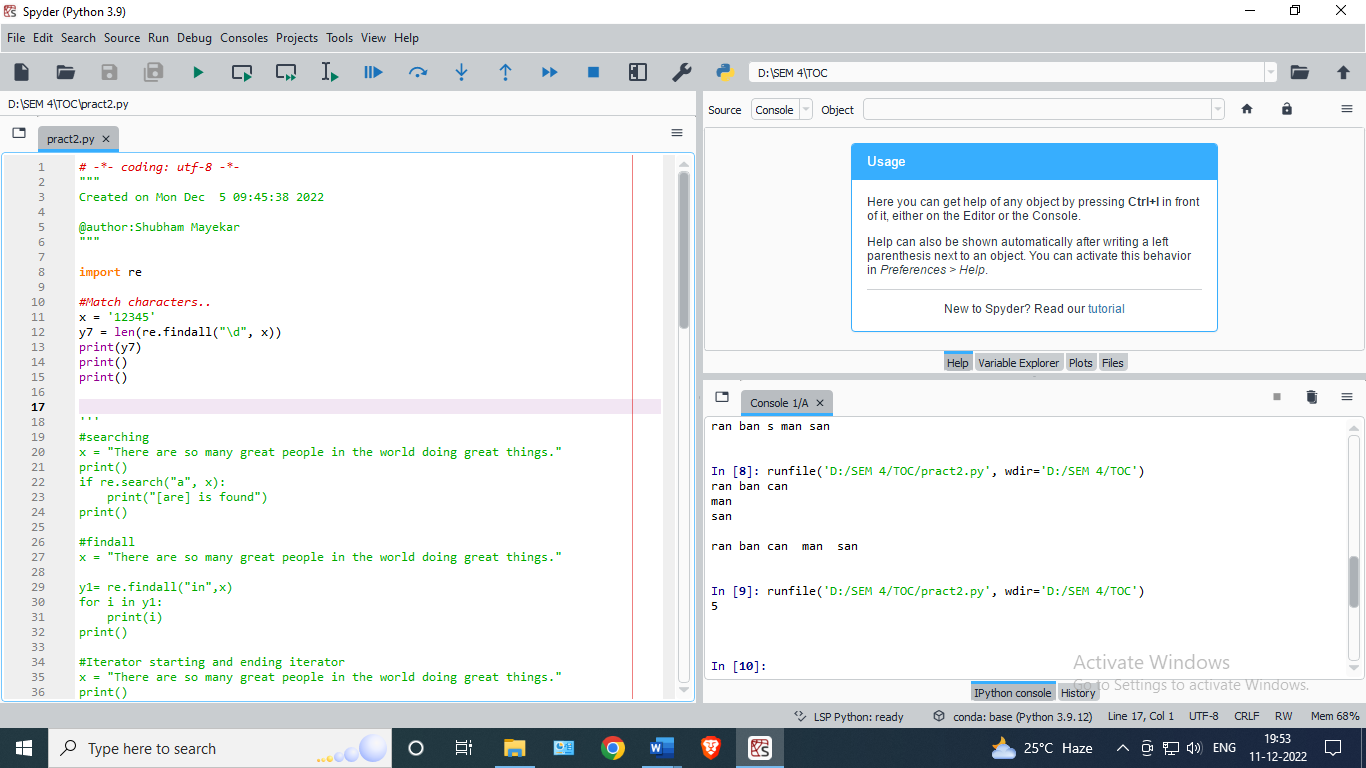
* **Output:**



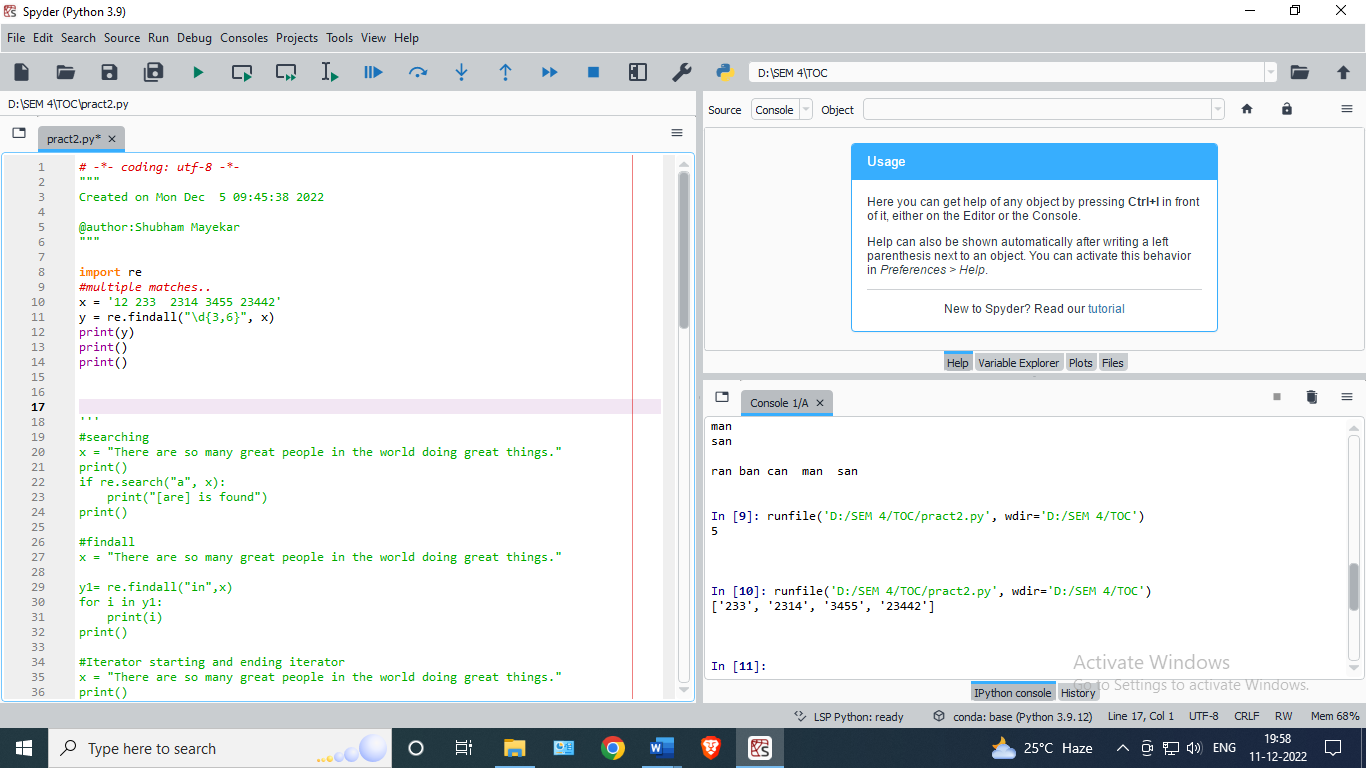
* **Match character :**



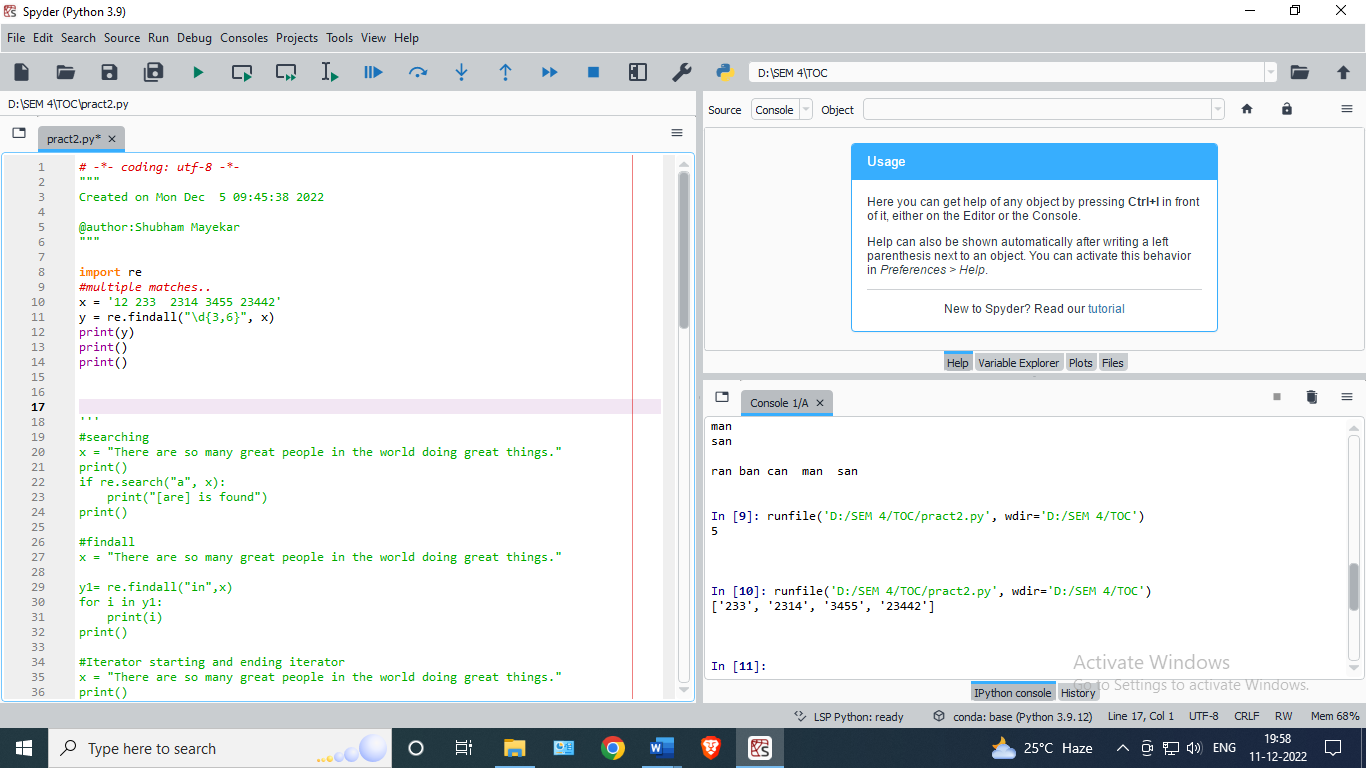
* **Output:**



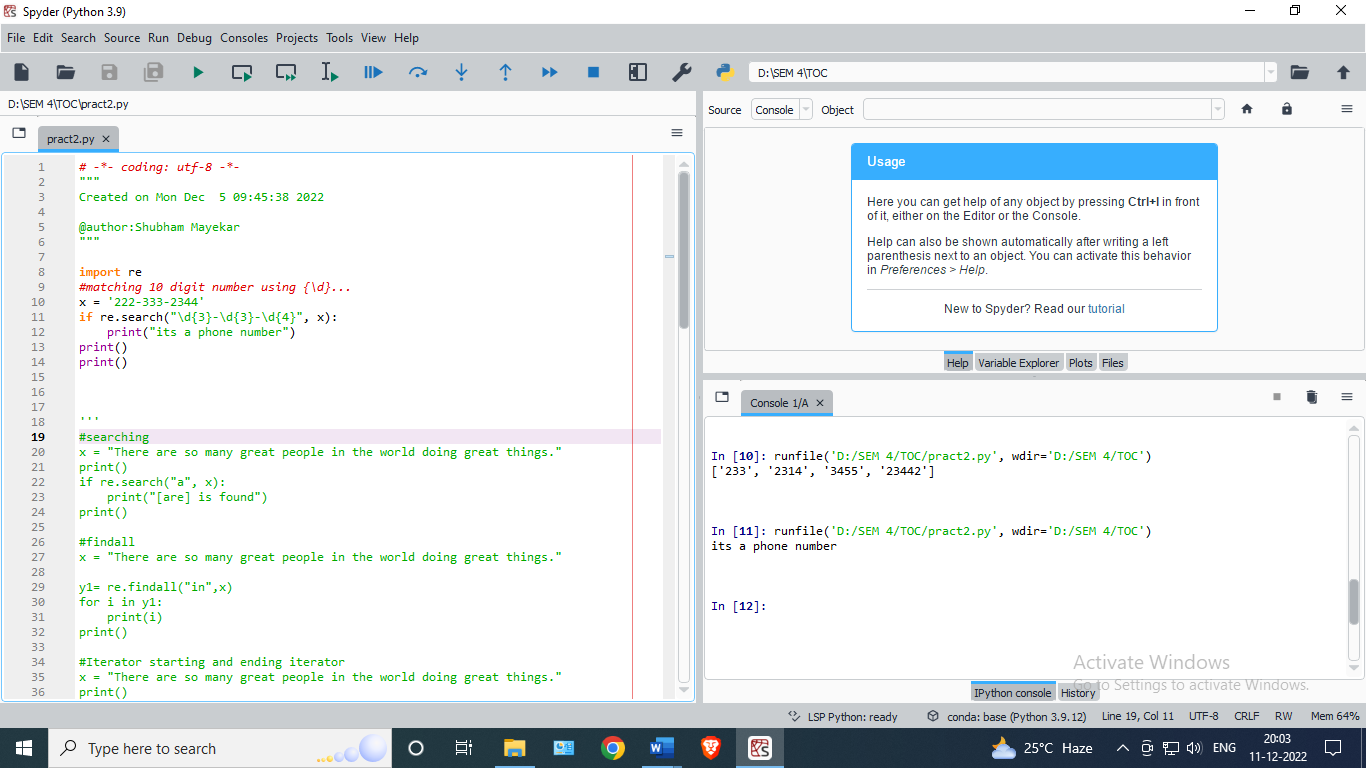
* **Multiple matches :**



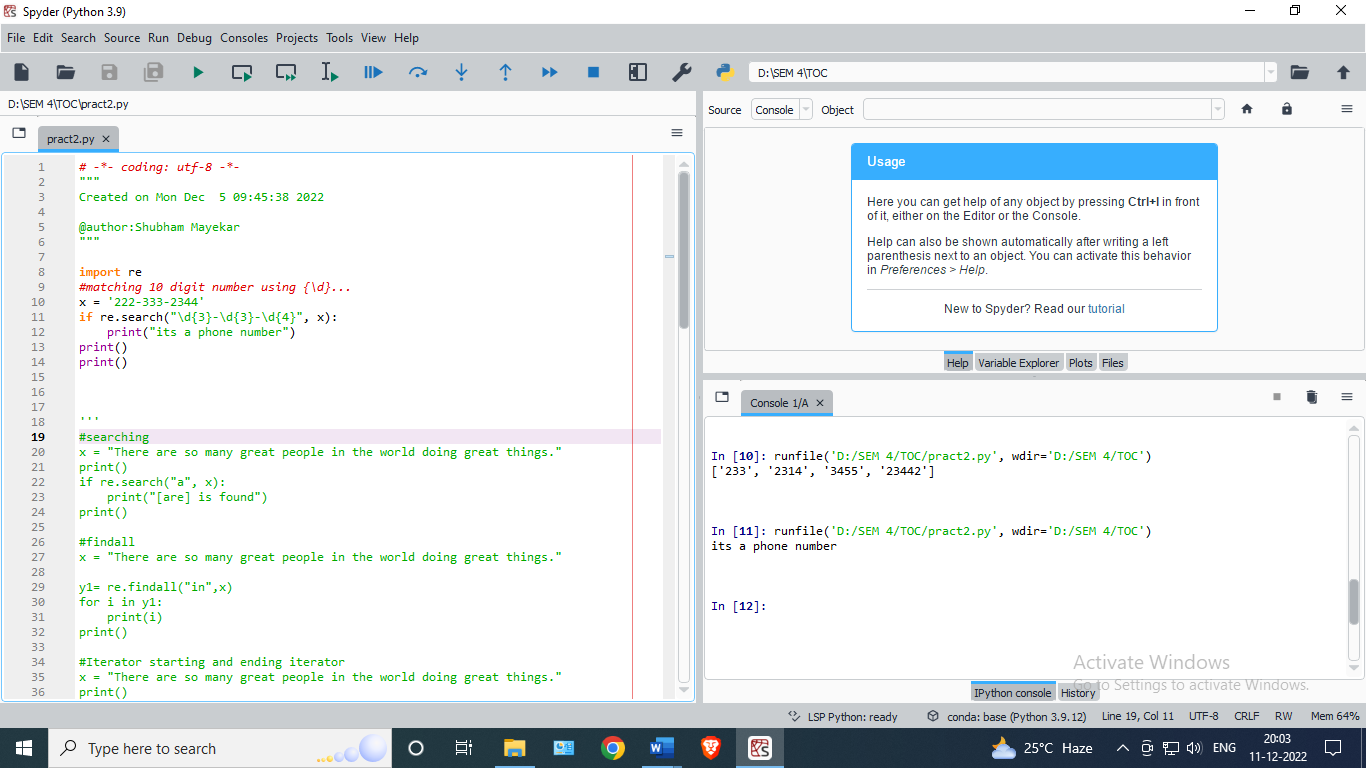
* **Output:**



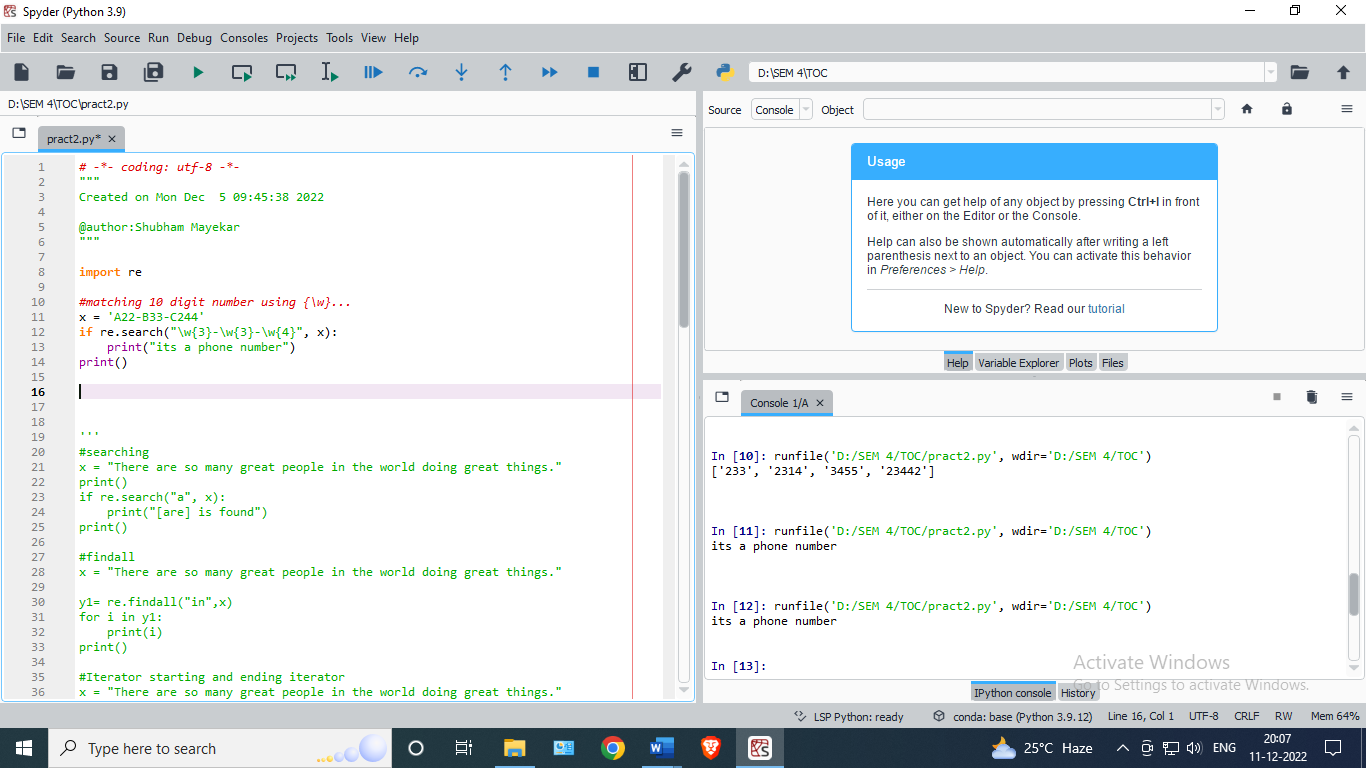
* **matching 10 digit number using {\d}:**



* **Output:**



* **matching 10 digit number using {\w}:**



* **Output:**

