STUDENT PORTFOLIO

Name: ANAS AHMED ATHER
Register Number: RA2011031010006
Mail ID: aa0094@srmist.edu.in
Department: NWC CSE
Specialization: Information technology
Semester: 5th

Subject Title: 18CSC301J Formal Language & Automata

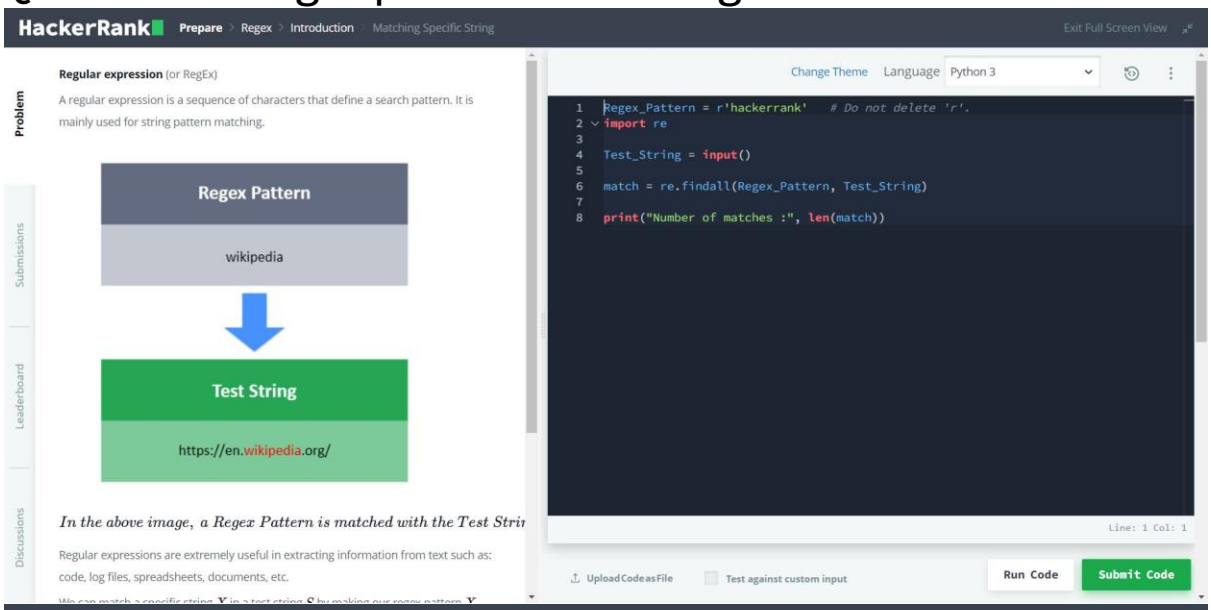
Handled By: Dr.P.Balaji Srikaanth (102776)

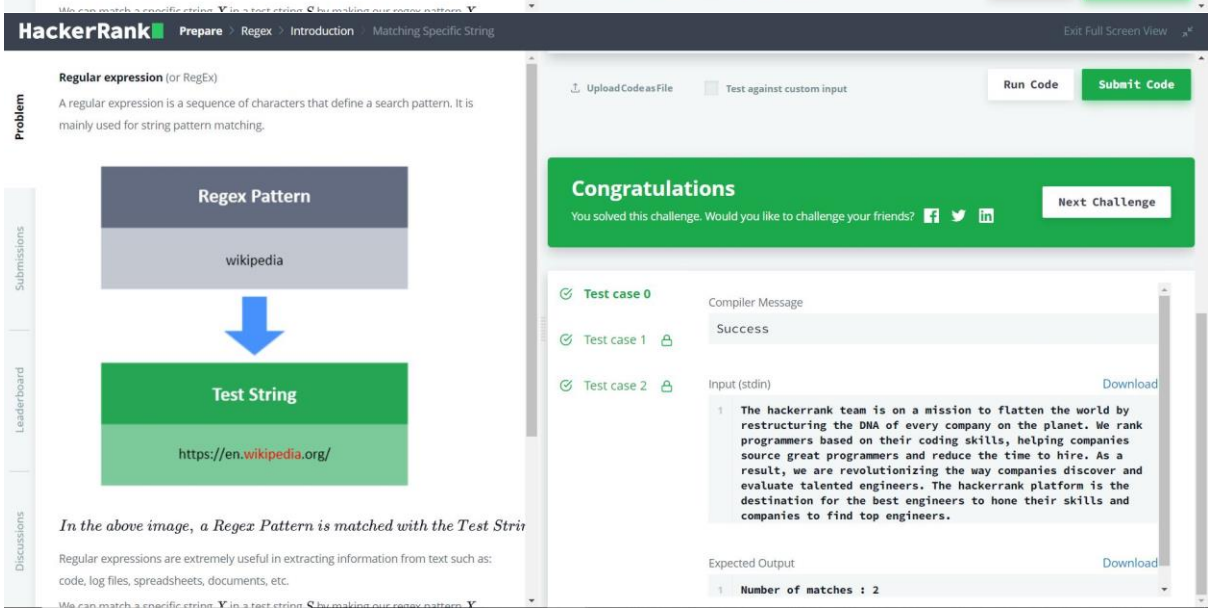
Regex Module of Hackerrank

SUBDOMAINS

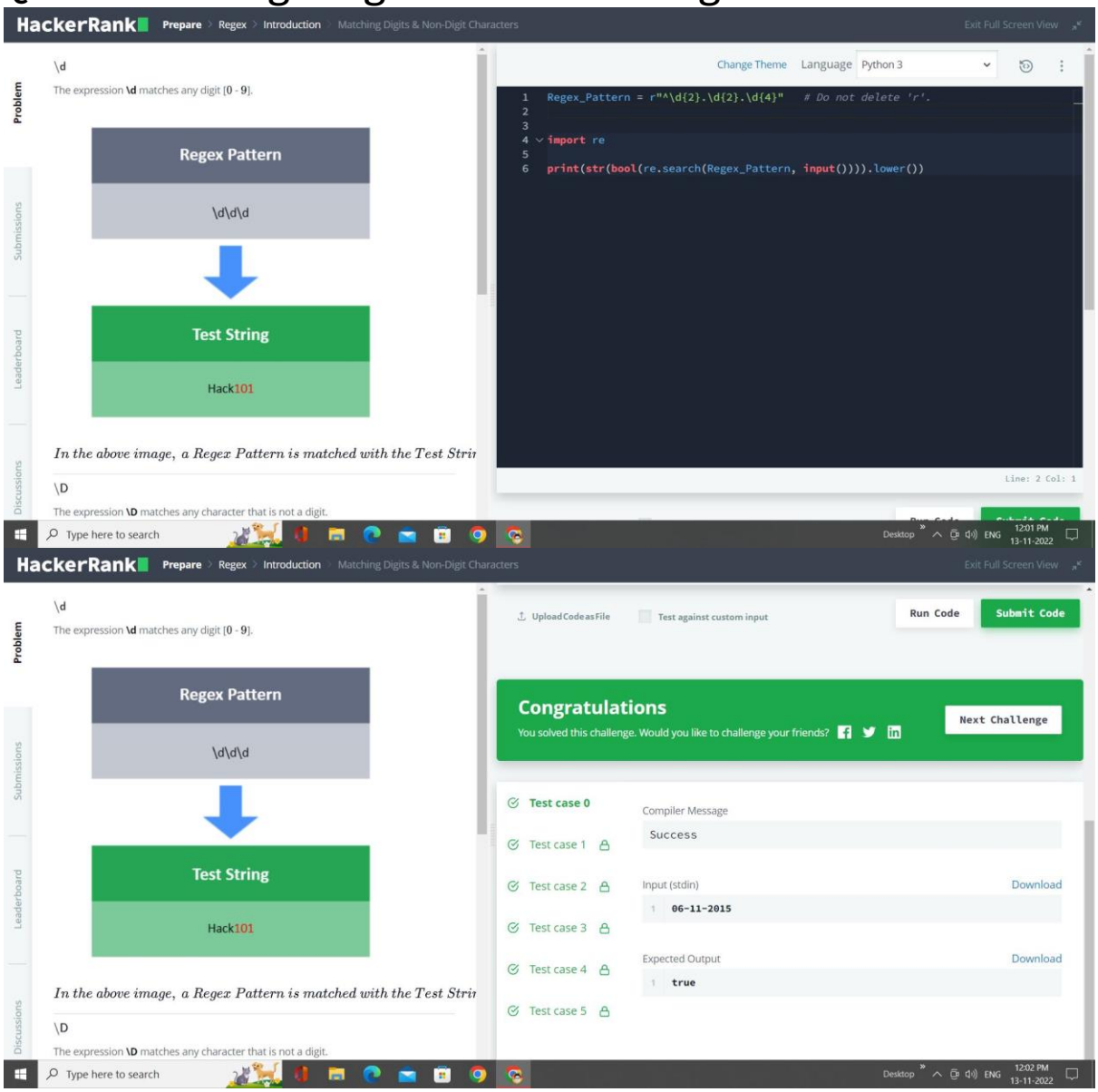
- [x] Introduction

Q.1.Matching Specific String

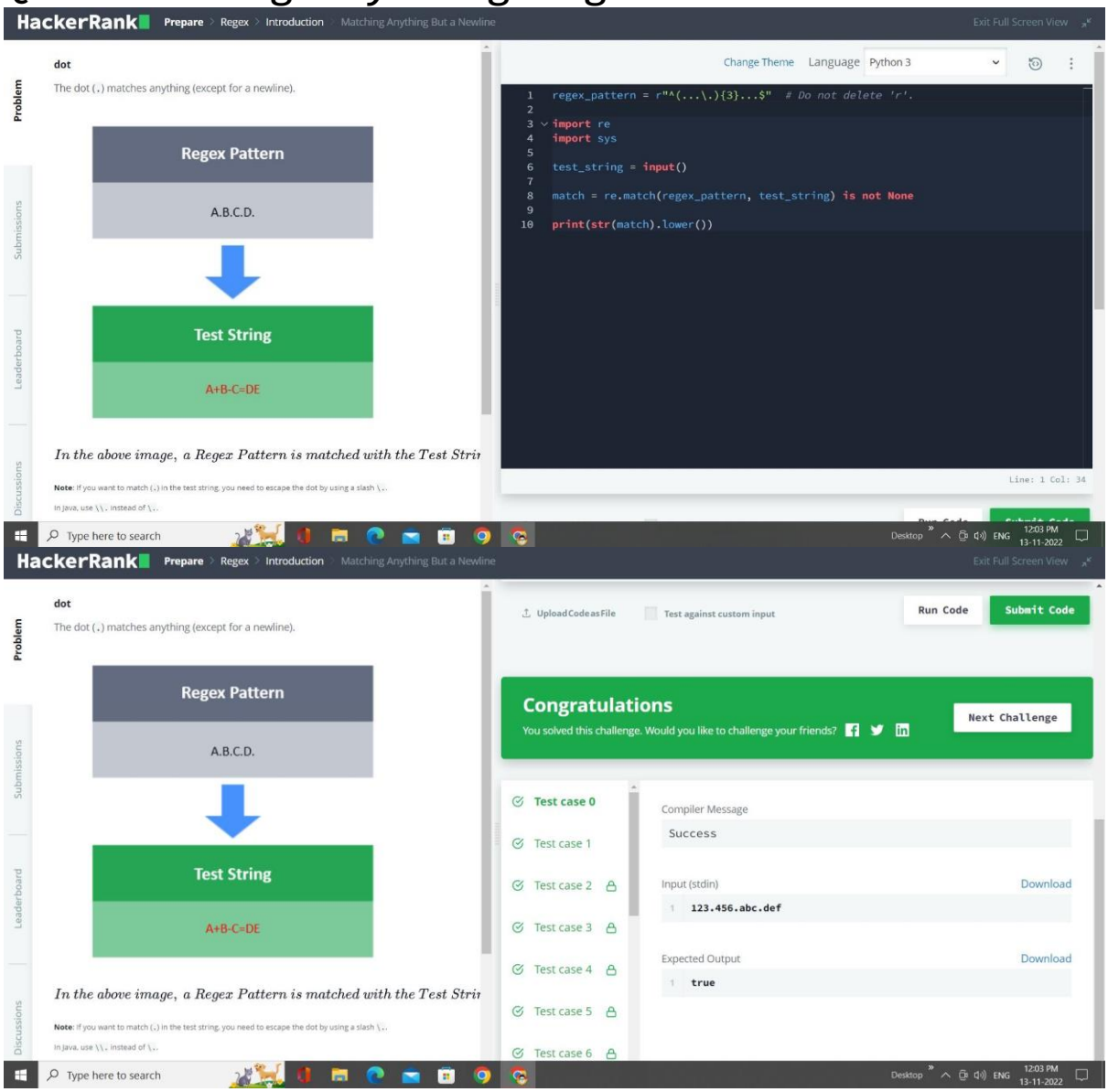




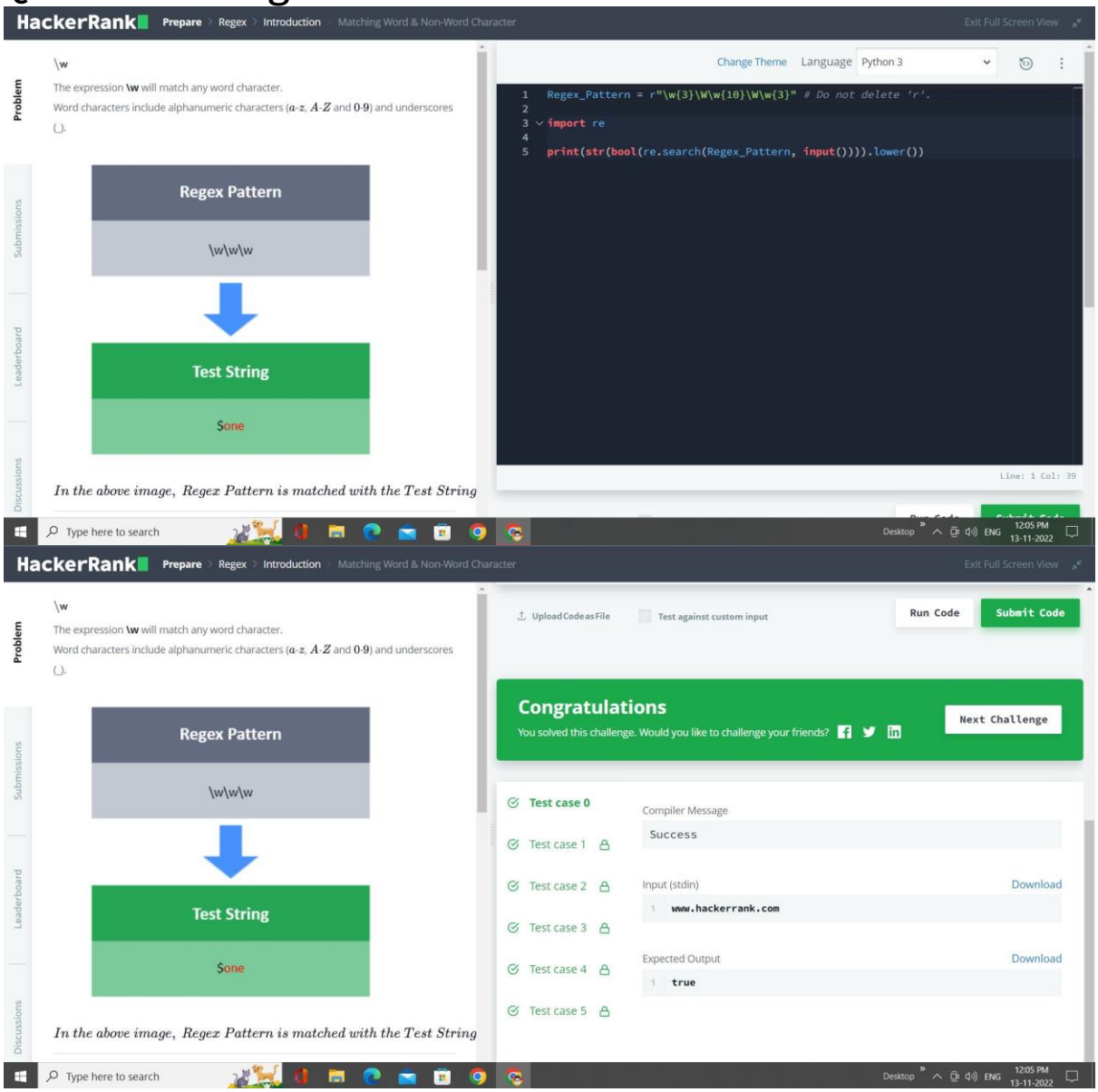
Q.2.Matching digit and non digit character

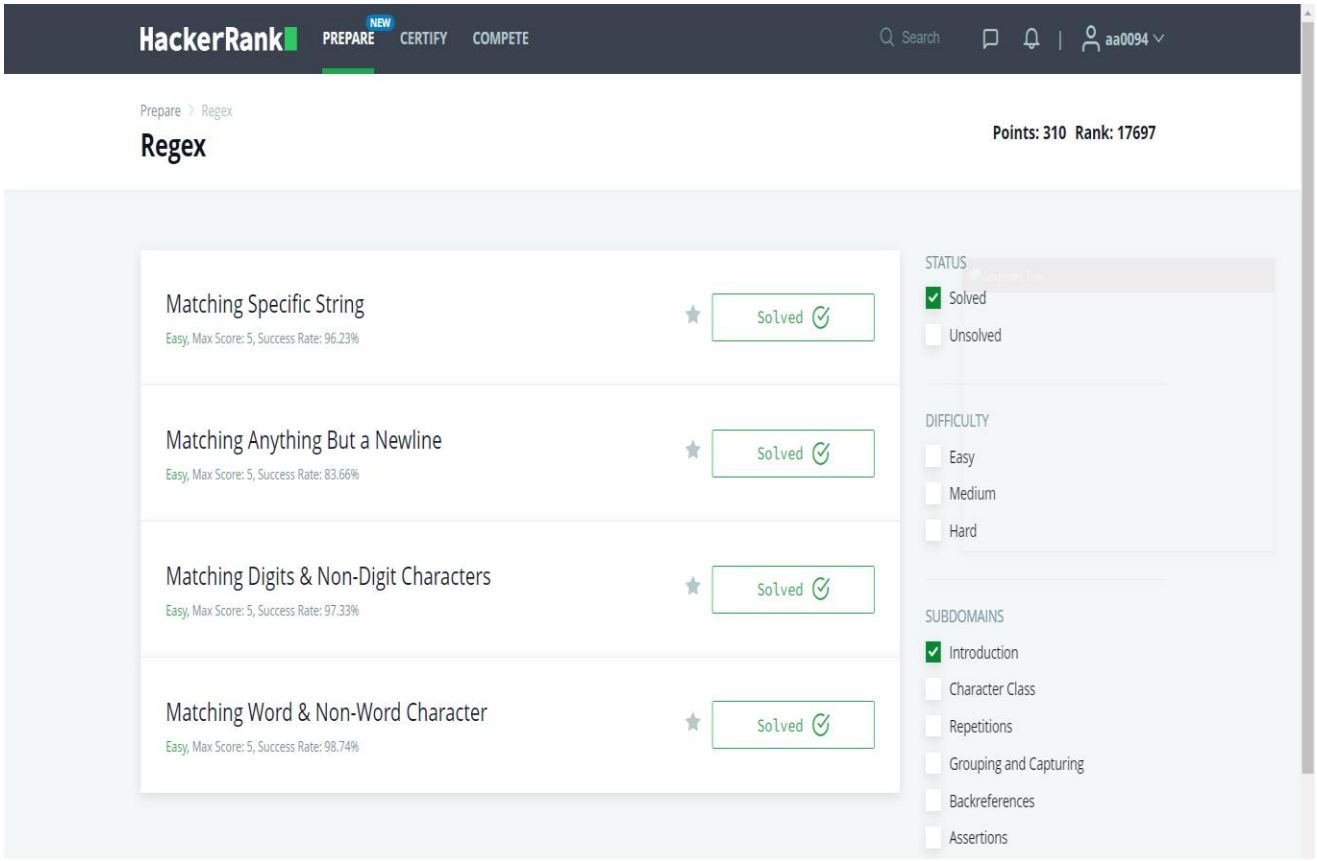


Q.3.Matching anything bug a newline



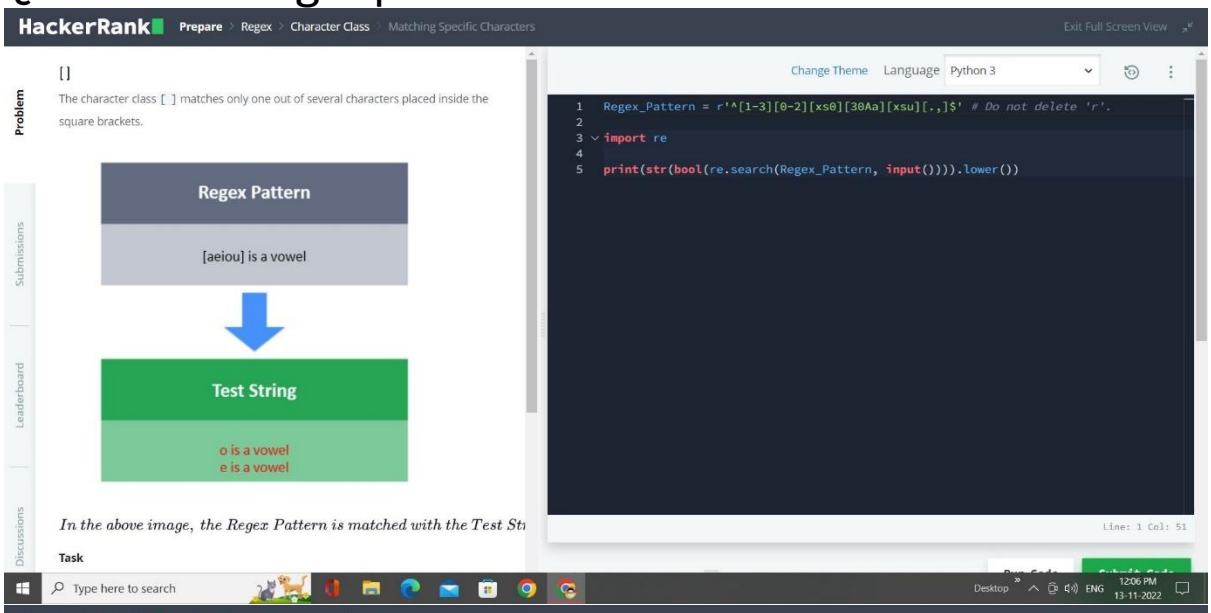
Q.4.Matching Word and non word character

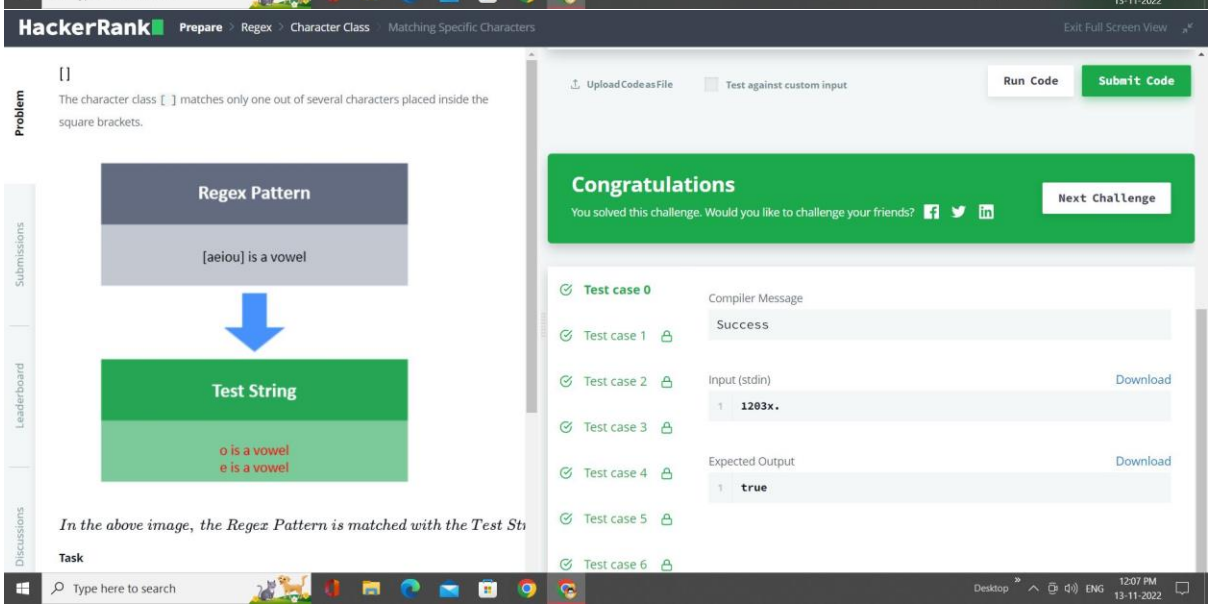



HackerRank
PREPARE
NEW
CERTIFY
COMPETE
Search
aa0094
Prepare > Regex
Regex
Points: 310 Rank: 17697
Matching Specific String
Easy, Max Score: 5, Success Rate: 96.23%
Solved
Matching Anything But a Newline
Easy, Max Score: 5, Success Rate: 83.66%
Solved
Matching Digits & Non-Digit Characters
Easy, Max Score: 5, Success Rate: 97.33%
Solved
Matching Word & Non-Word Character
Easy, Max Score: 5, Success Rate: 98.74%
Solved
STATUS
Solved
Unsolved
DIFFICULTY
Easy
Medium
Hard
SUBDOMAINS
Introduction
Character Class
Repetitions
Grouping and Capturing
Backreferences
Assertions

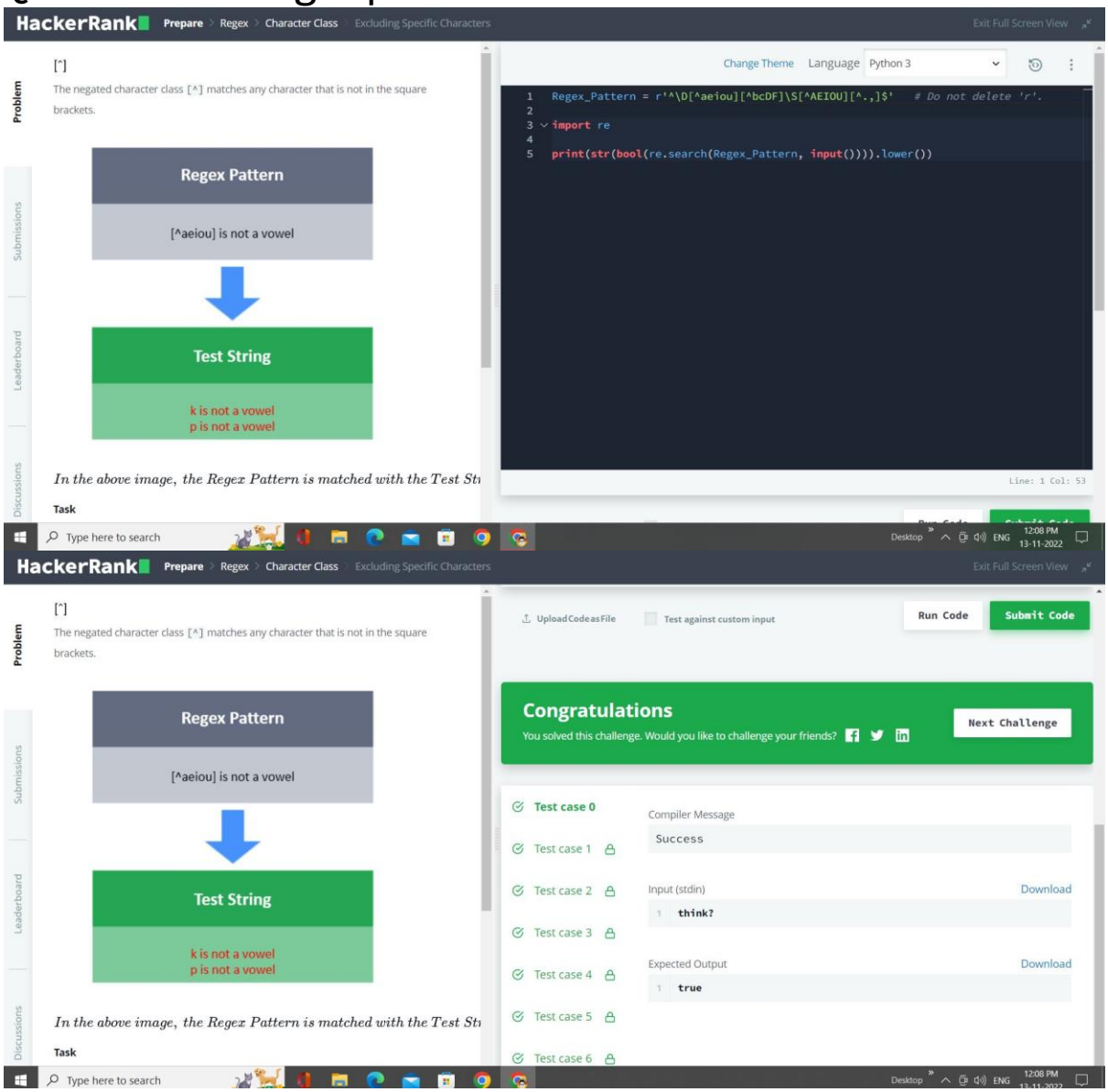
☑ Character Class

Q.5.Matching Specific character

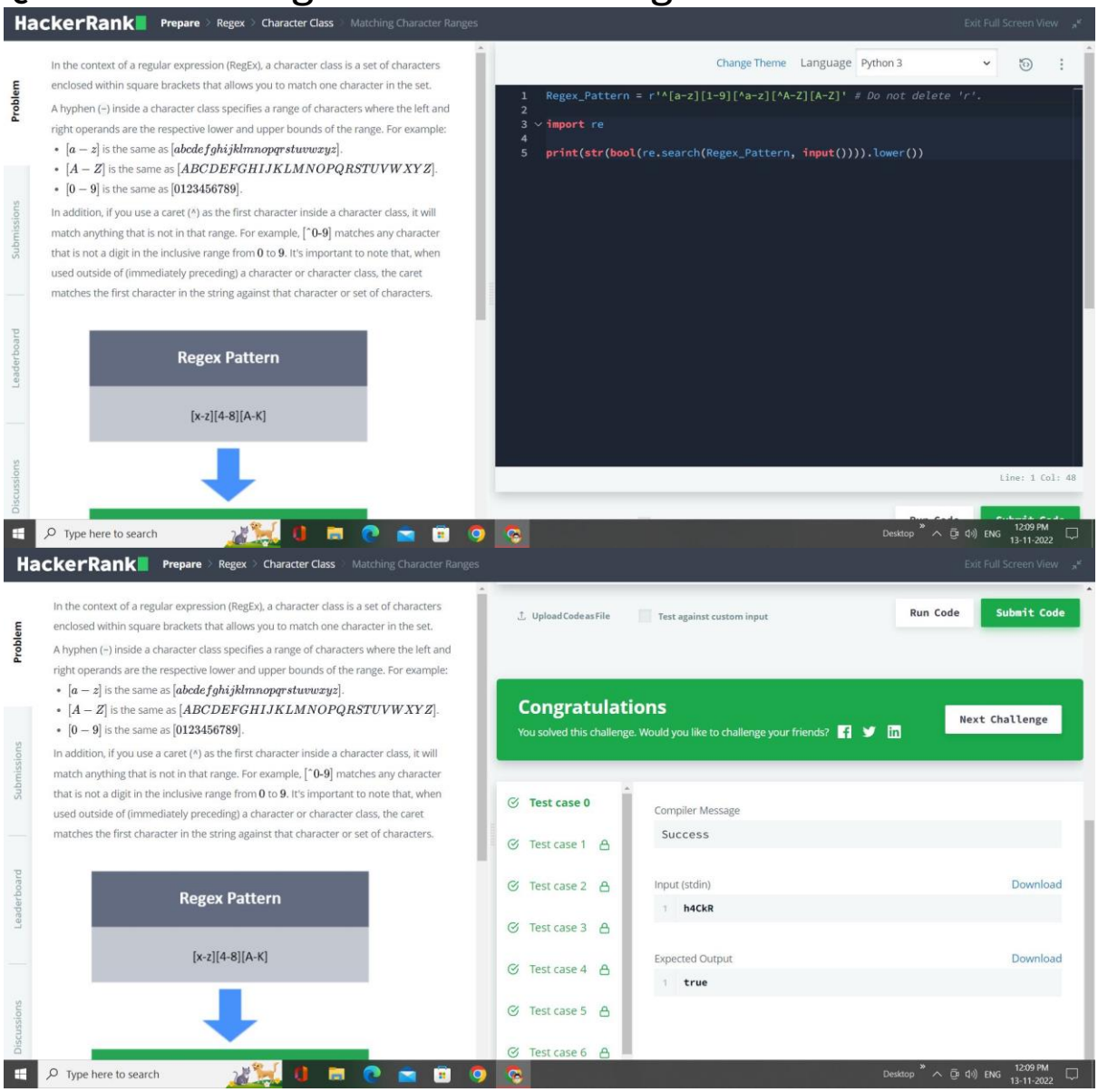


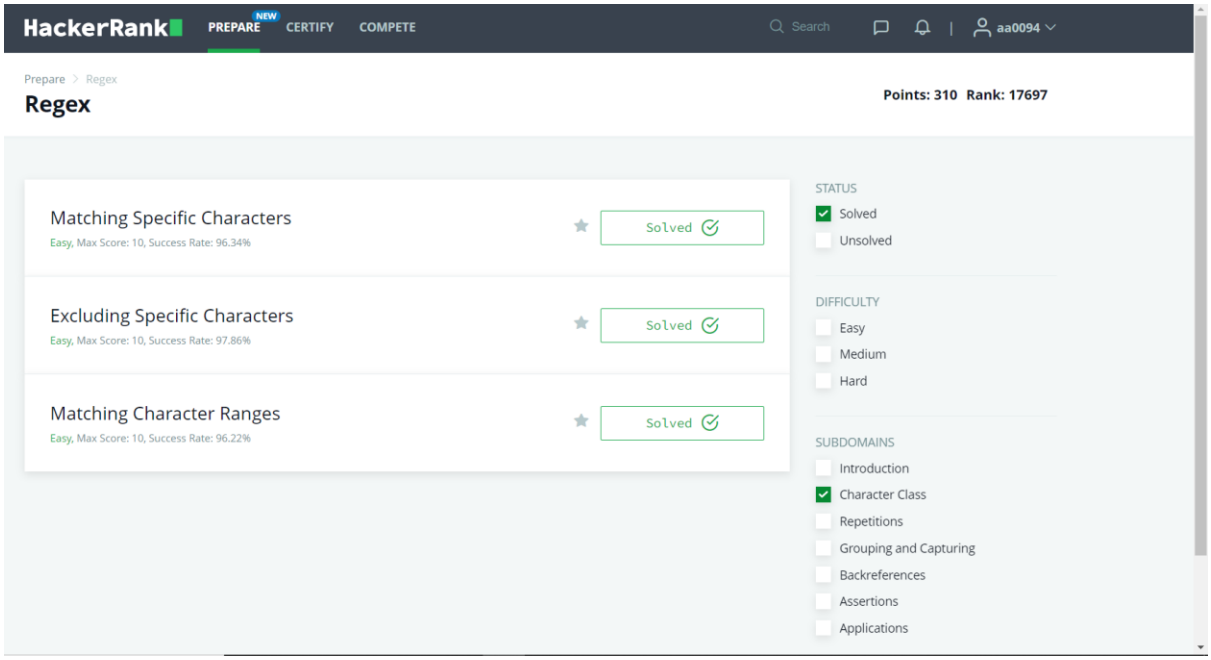


Q.6.Excluding specific character



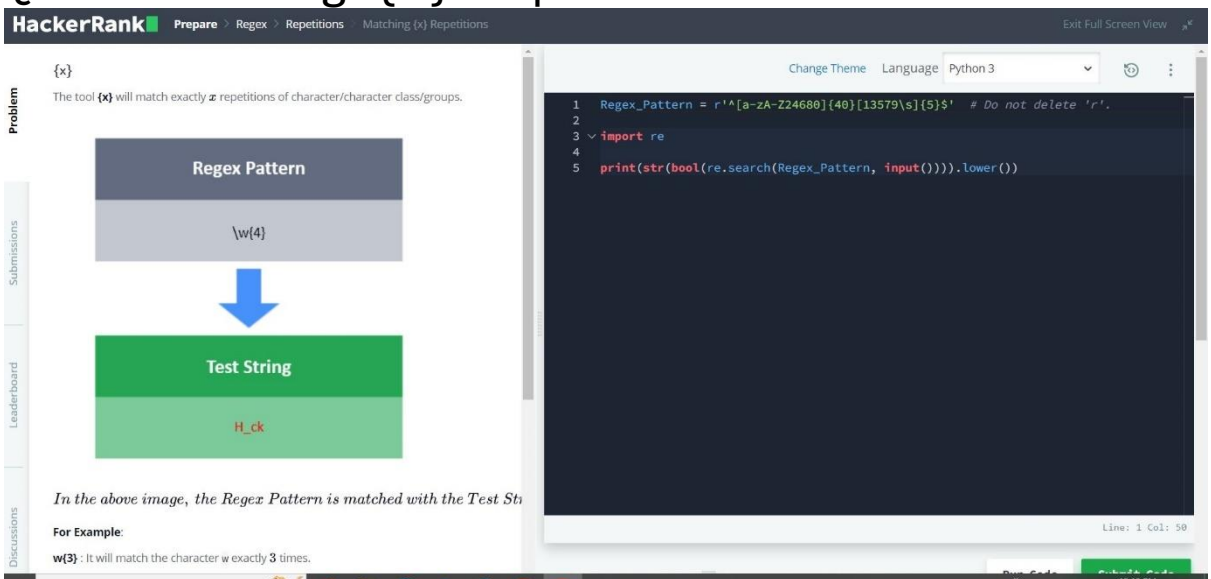
Q.7. Matching character ranges

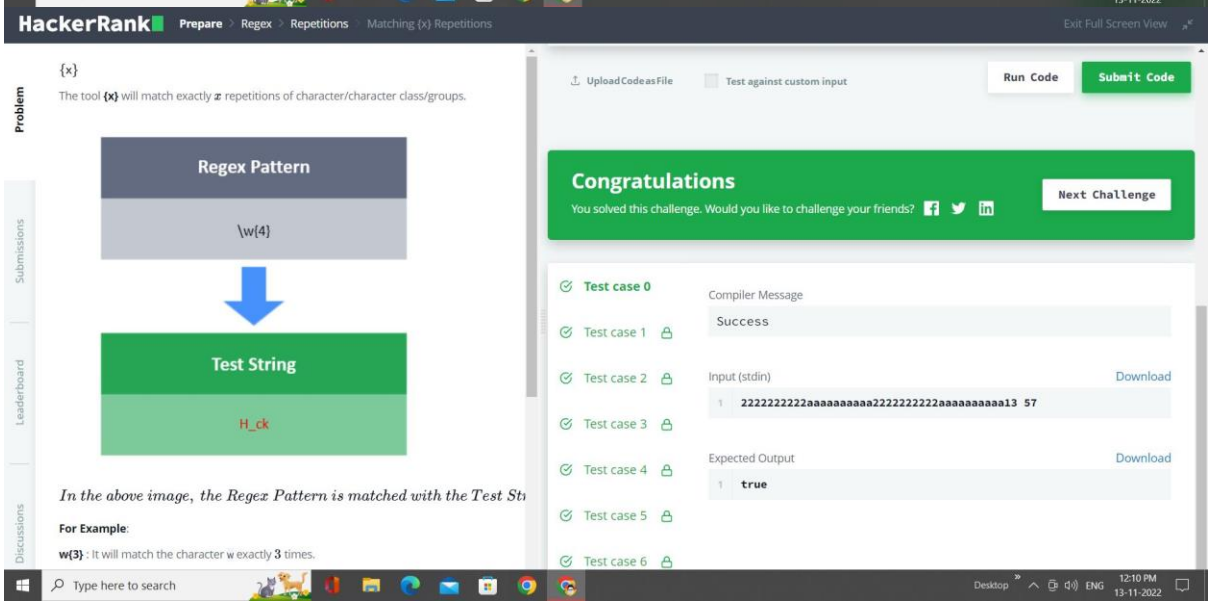



HackerRank
PREPARE
NEW
CERTIFY
COMPETE
Search
aa0094
Prepare > Regex
Regex
Points: 310 Rank: 17697
Matching Specific Characters
Easy, Max Score: 10, Success Rate: 96.34%
Solved
Excluding Specific Characters
Easy, Max Score: 10, Success Rate: 97.86%
Solved
Matching Character Ranges
Easy, Max Score: 10, Success Rate: 96.22%
Solved
STATUS
Solved
Unsolved
DIFFICULTY
Easy
Medium
Hard
SUBDOMAINS
Introduction
Character Class
Repetitions
Grouping and Capturing
Backreferences
Assertions
Applications

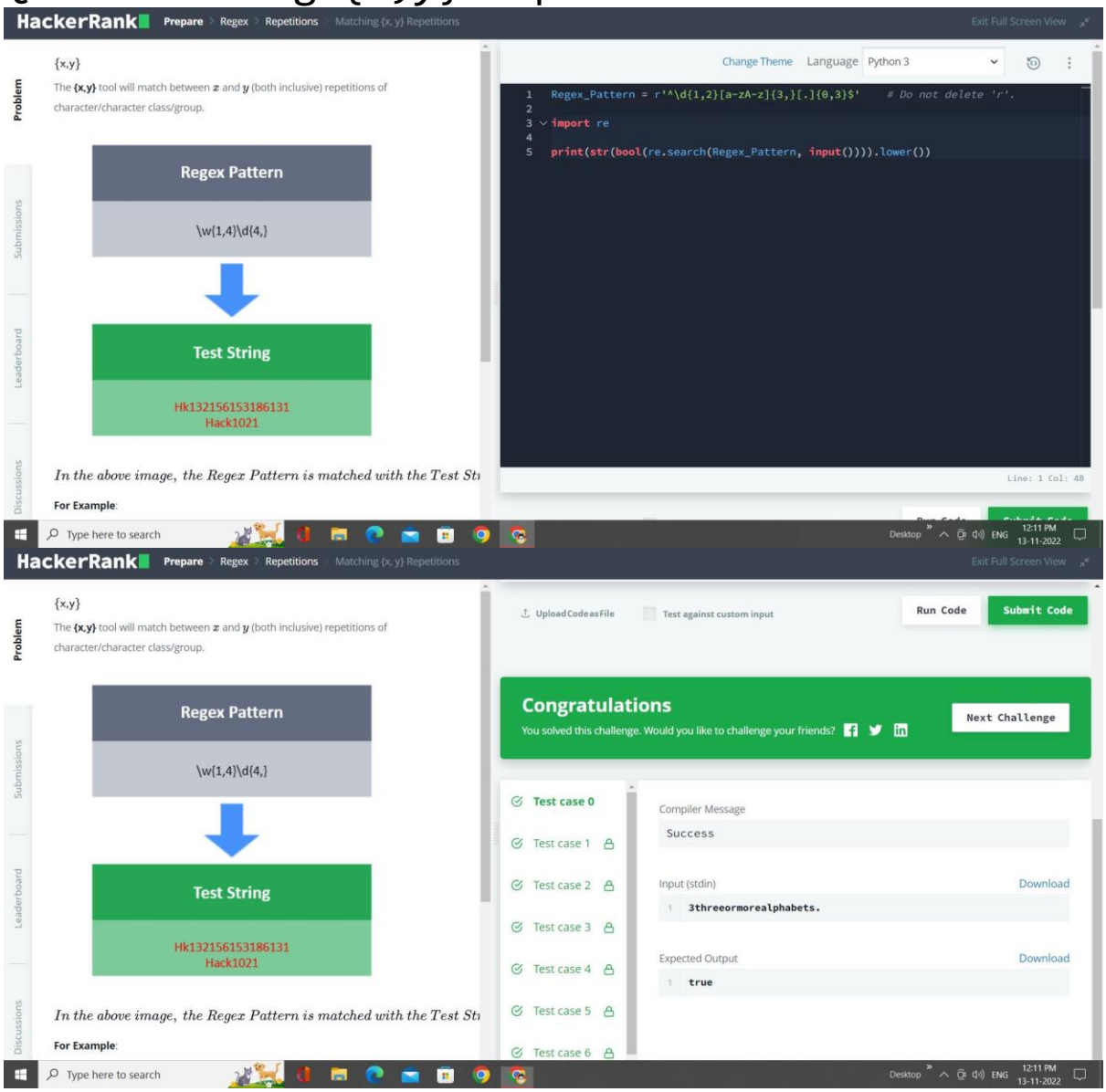
Repetitions

Q.8. Matching {x} repetition

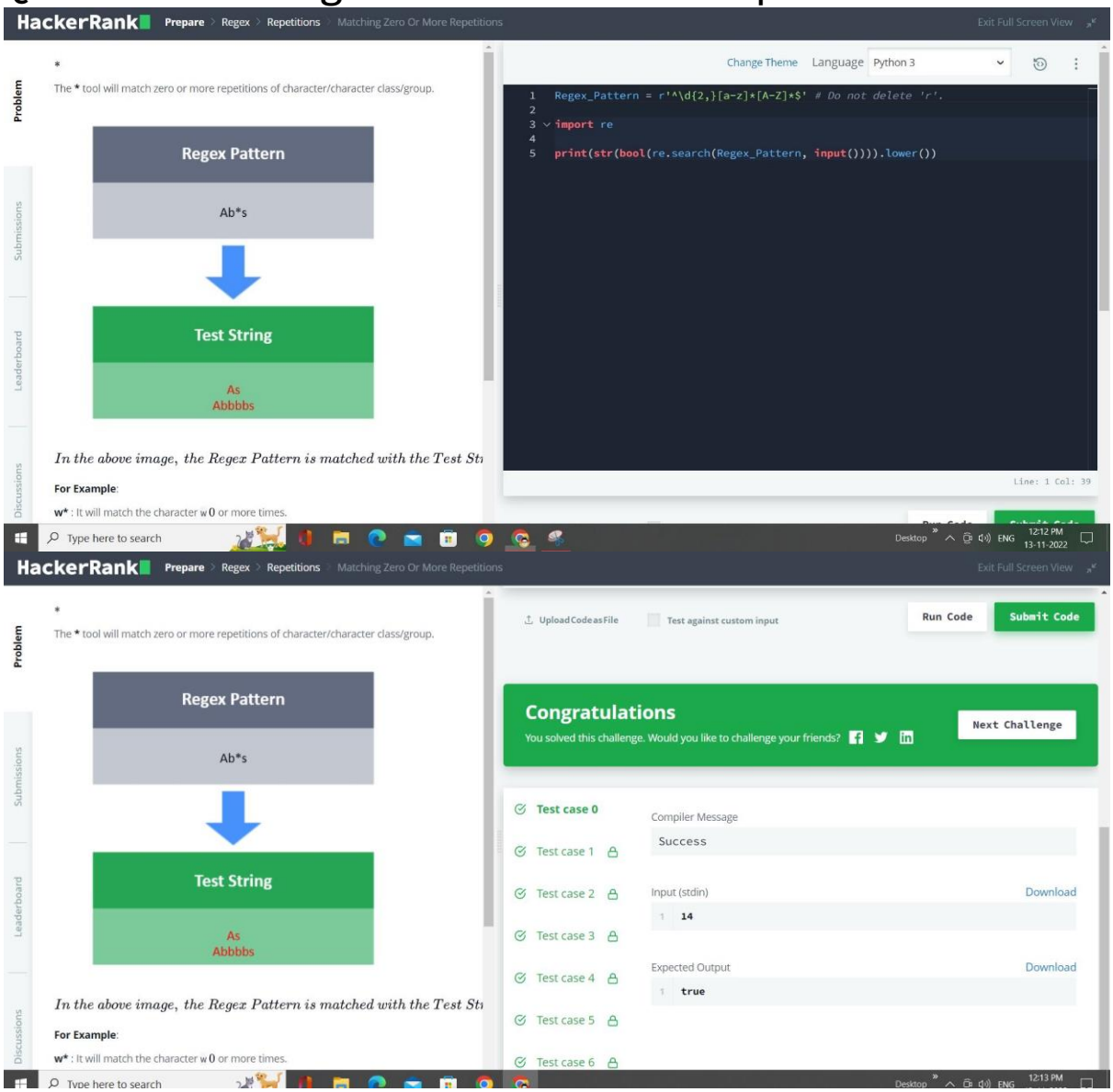


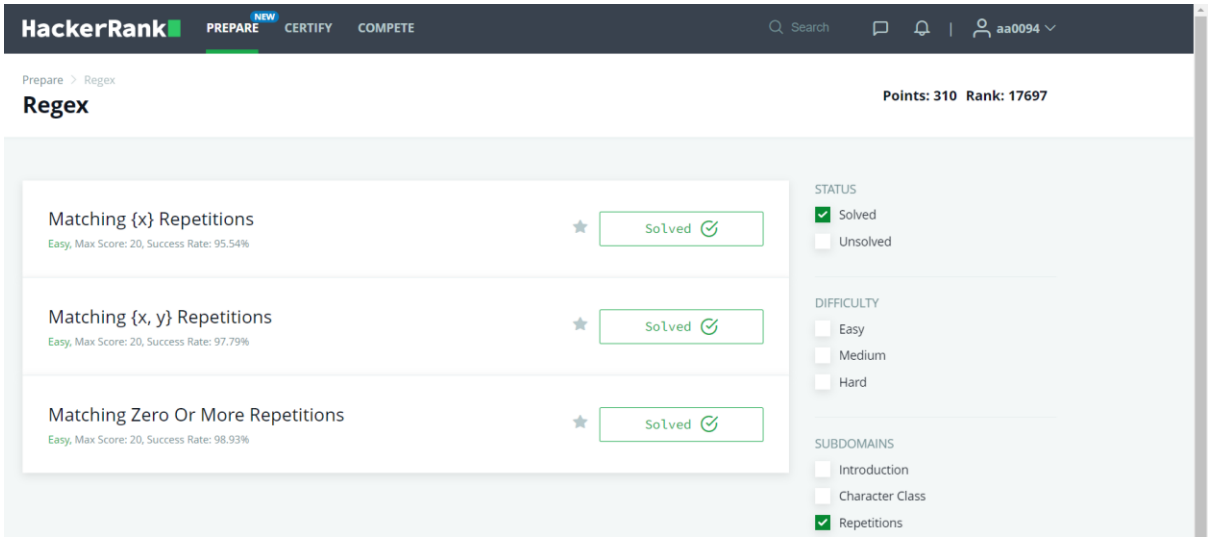


Q.9. Matching {x,y} Repititions



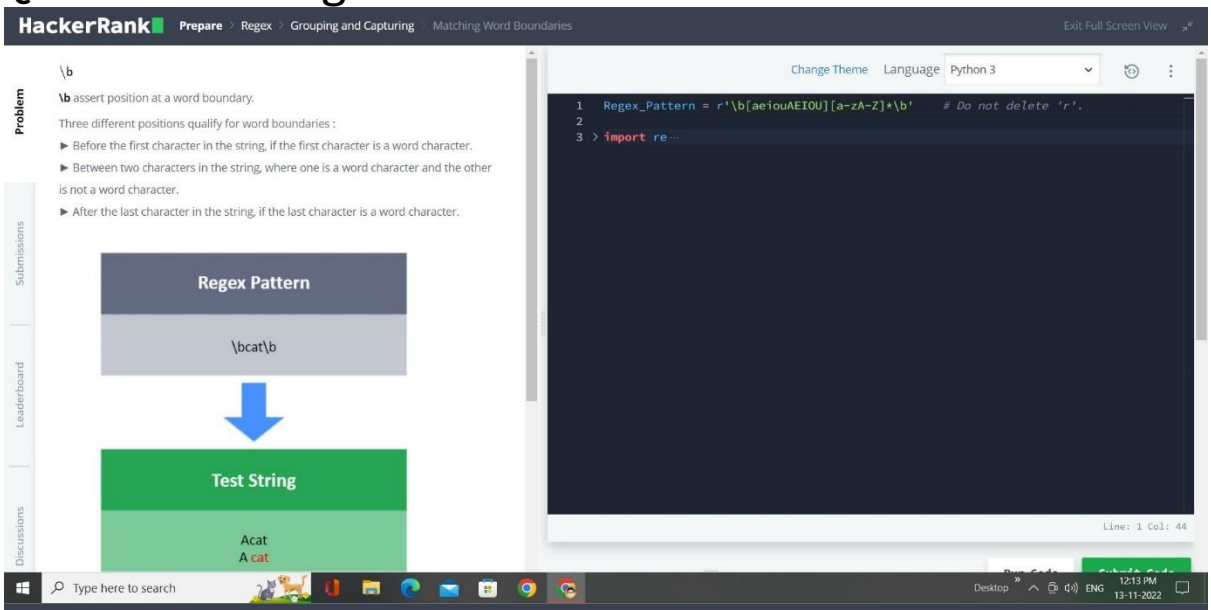
Q.10. Matching zero and more repetitions

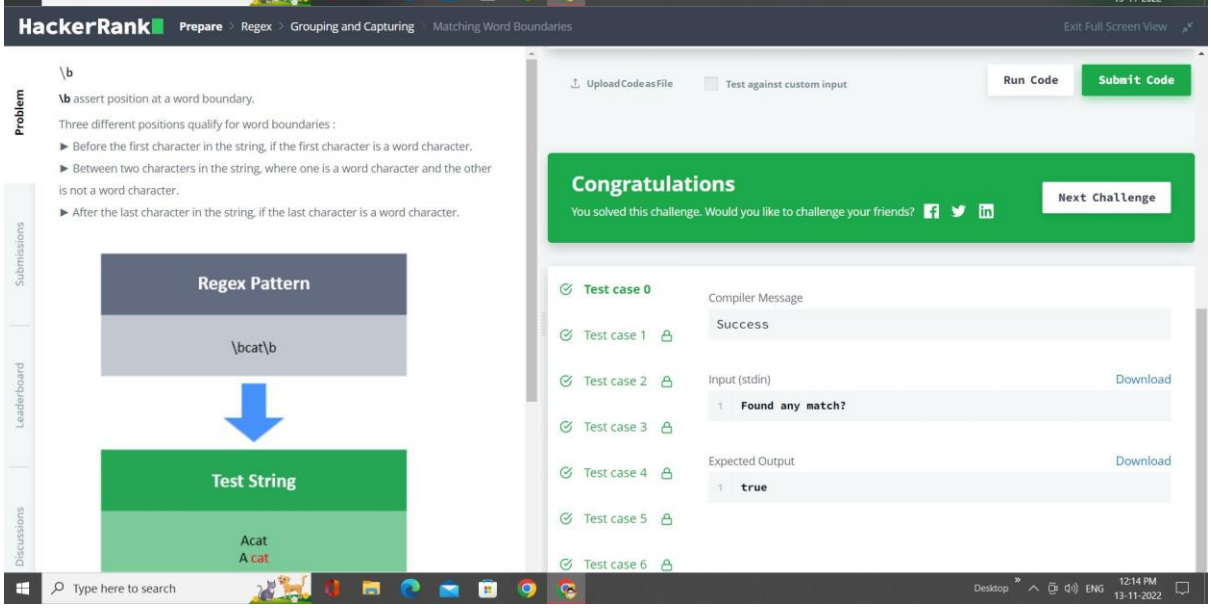



HackerRank
PREPARE
NEW
CERTIFY
COMPETE
Search
aa0094
Prepare > Regex
Regex
Points: 310 Rank: 17697
Matching {x} Repetitions
Easy, Max Score: 20, Success Rate: 95.54%
Solved
Matching {x, y} Repetitions
Easy, Max Score: 20, Success Rate: 97.79%
Solved
Matching Zero Or More Repetitions
Easy, Max Score: 20, Success Rate: 98.93%
Solved
STATUS
Solved
Unsolved
DIFFICULTY
Easy
Medium
Hard
SUBDOMAINS
Introduction
Character Class
Repetitions

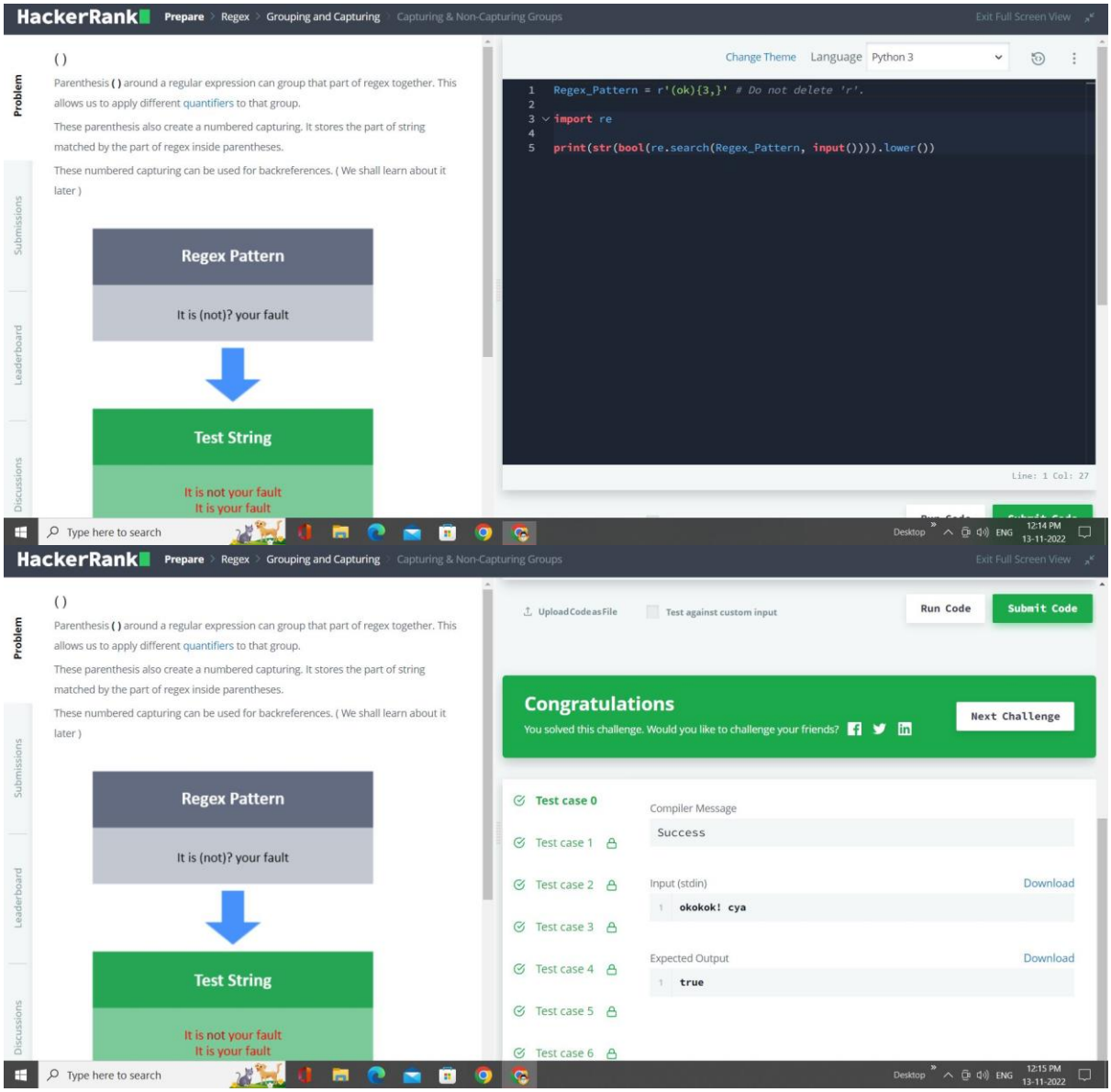
Grouping and Capturing

Q.11.Matching word boundaries

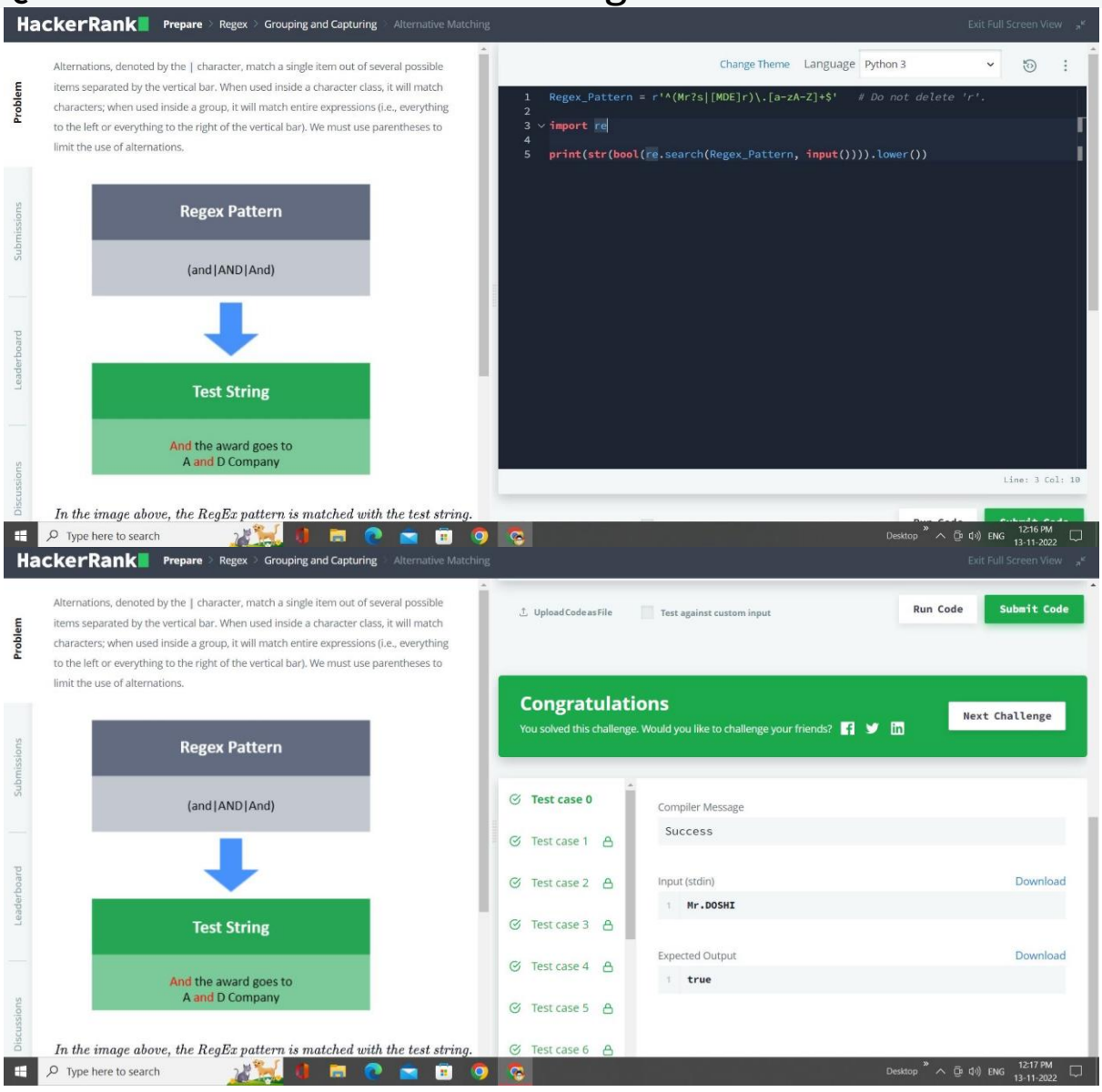


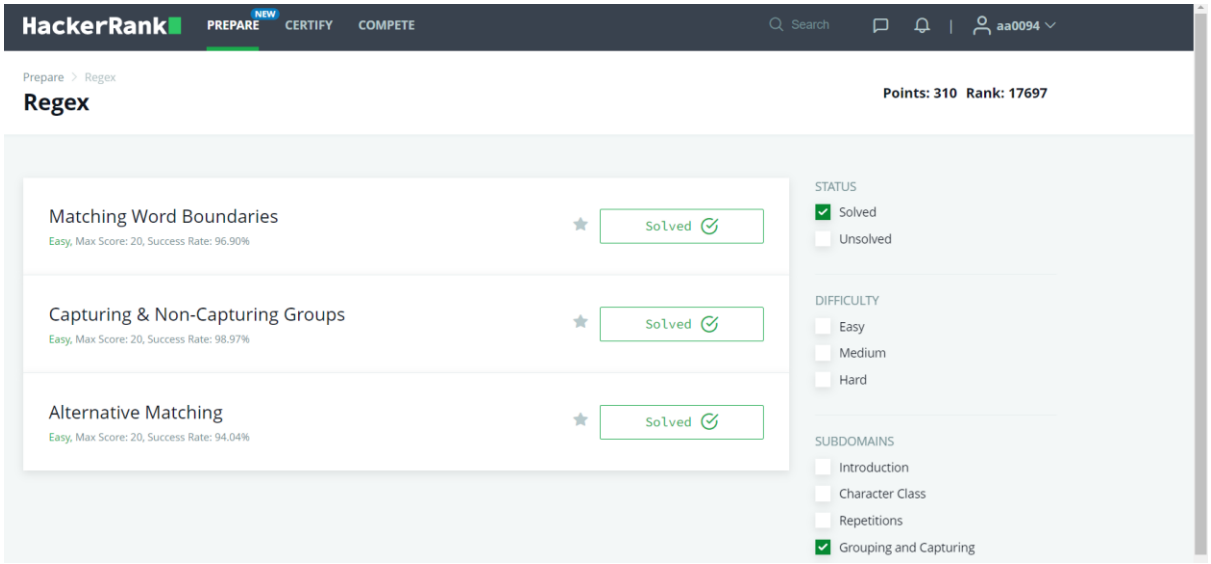


Q.12.Capturing and non-capturing group



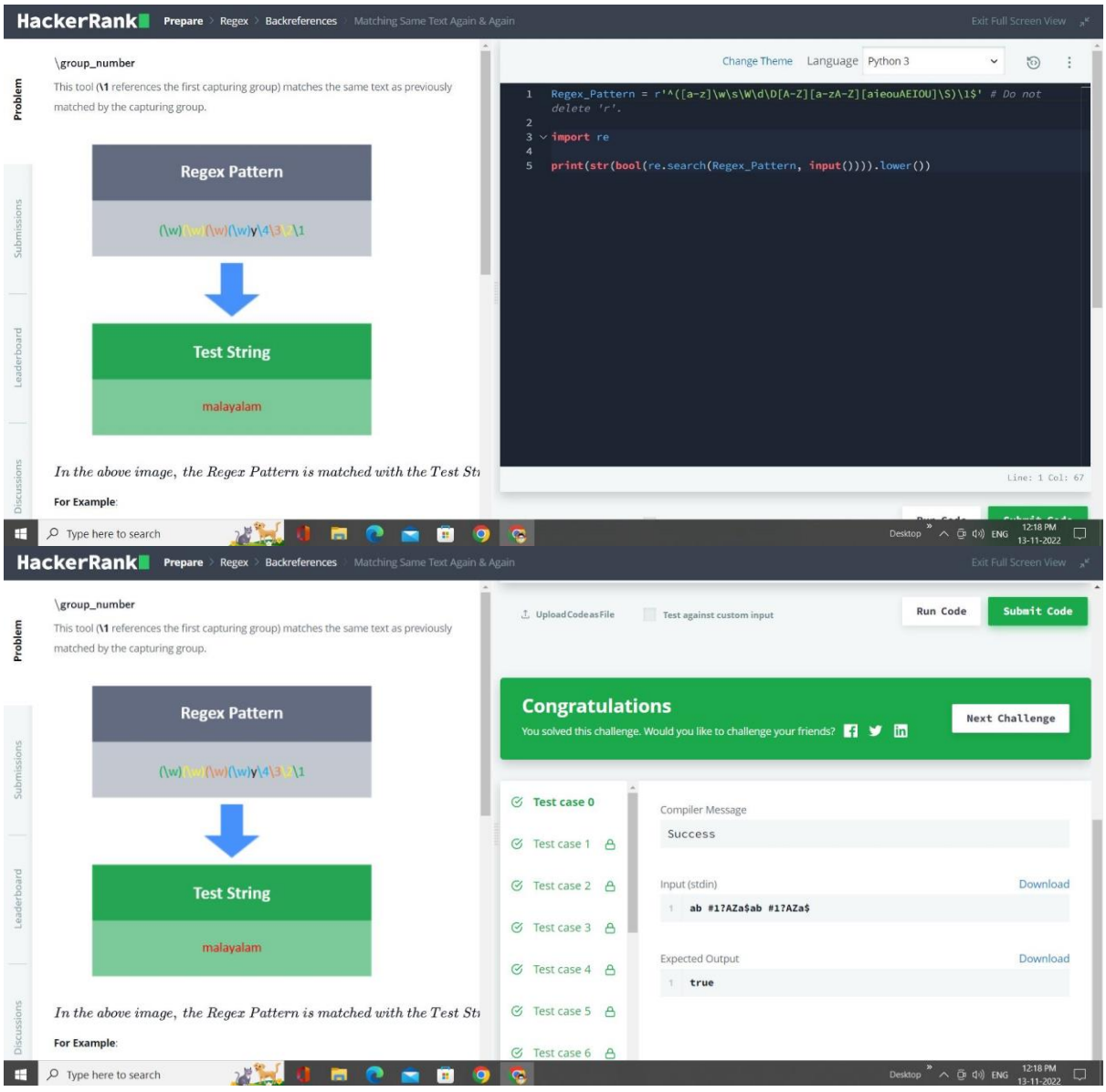
Q.13. Alternative Matching



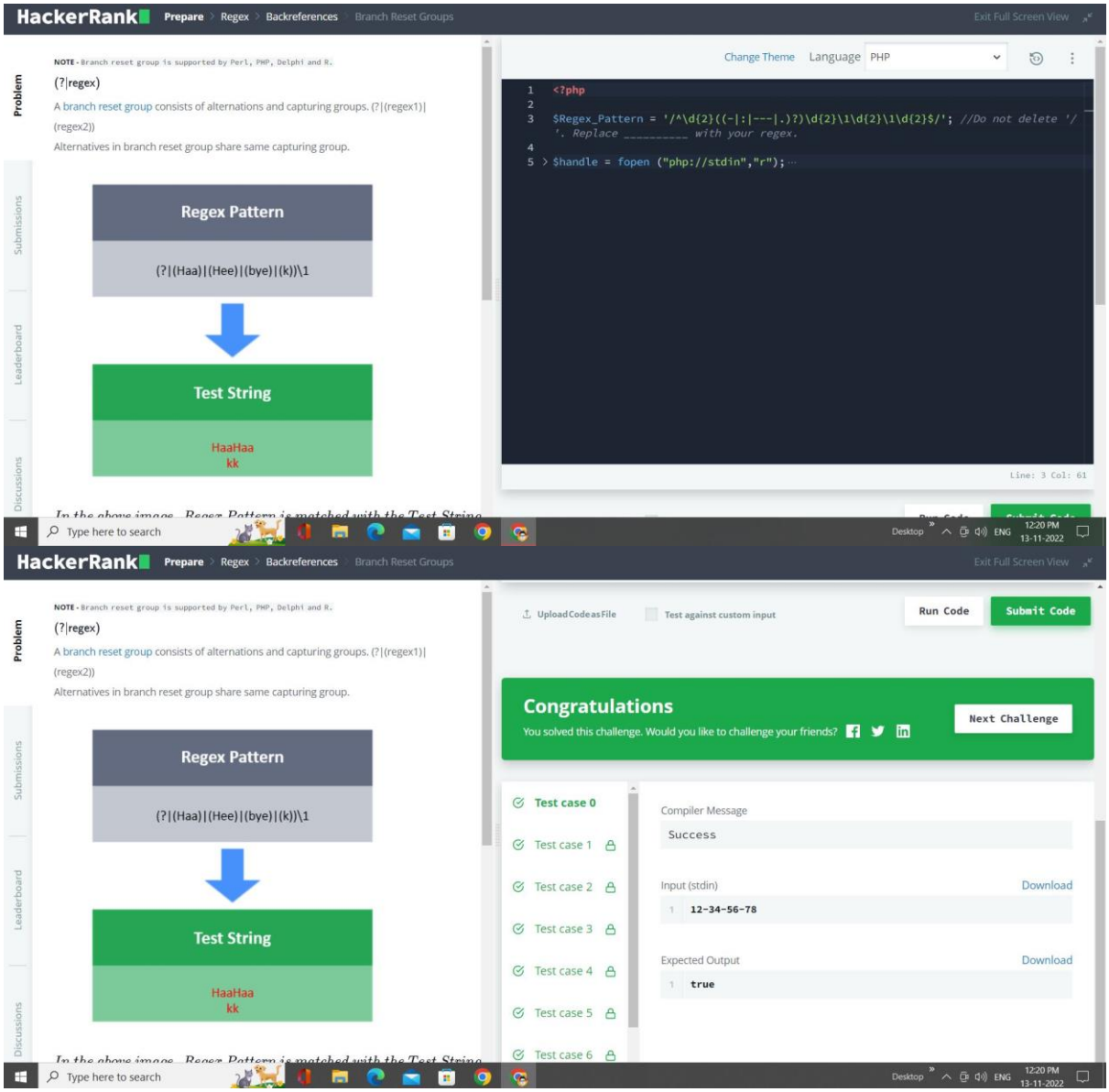

HackerRank
PREPARE
NEW
CERTIFY
COMPETE
Search
aa0094
Prepare
Regex
Regex
Points: 310 Rank: 17697
Matching Word Boundaries
Easy, Max Score: 20, Success Rate: 96.90%
Solved
Capturing & Non-Capturing Groups
Easy, Max Score: 20, Success Rate: 98.97%
Solved
Alternative Matching
Easy, Max Score: 20, Success Rate: 94.04%
Solved
STATUS
Solved
Unsolved
DIFFICULTY
Easy
Medium
Hard
SUBDOMAINS
Introduction
Character Class
Repetitions
Grouping and Capturing

☑ Backreferences

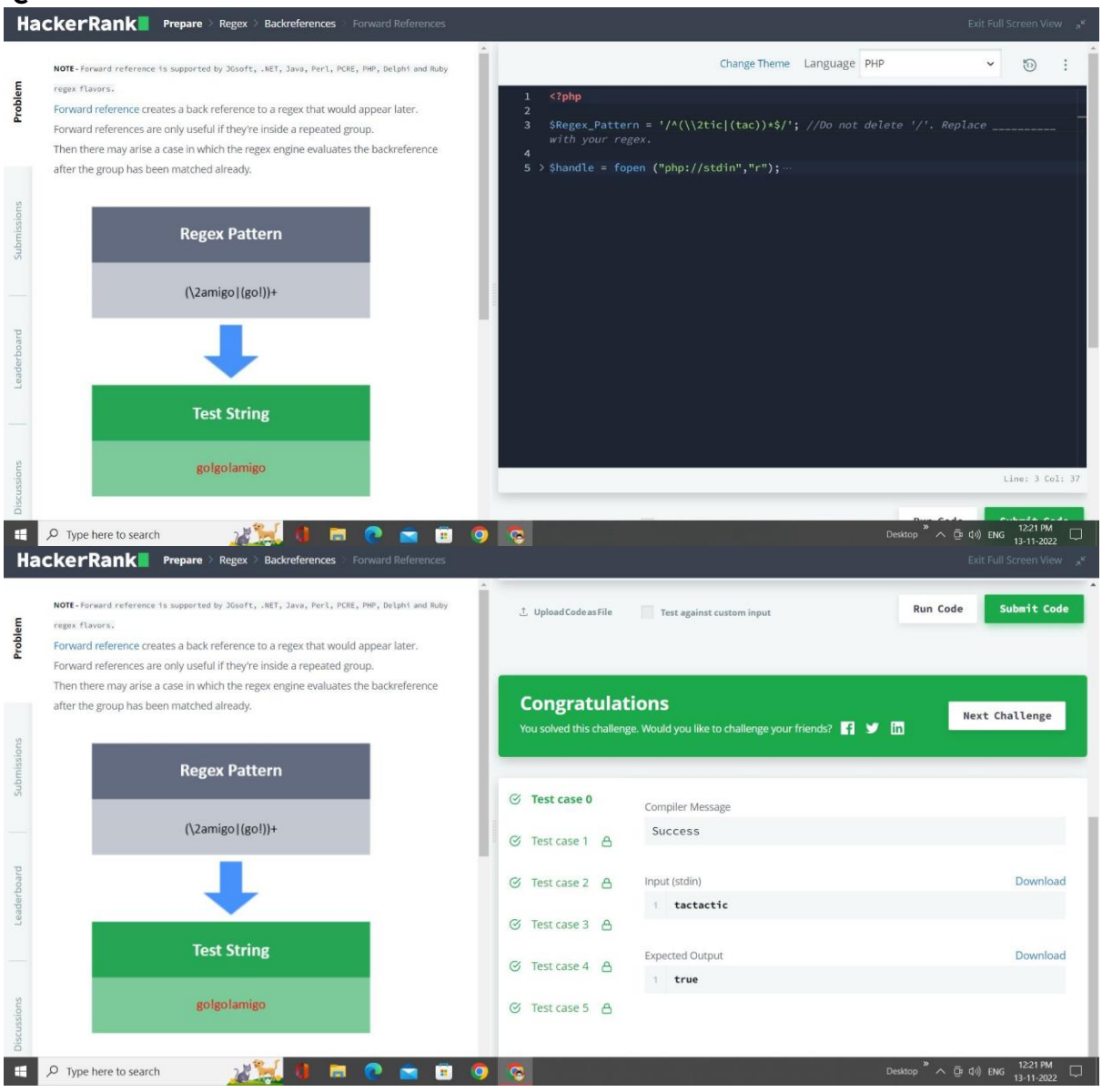
Q.14.Matching same text again and again

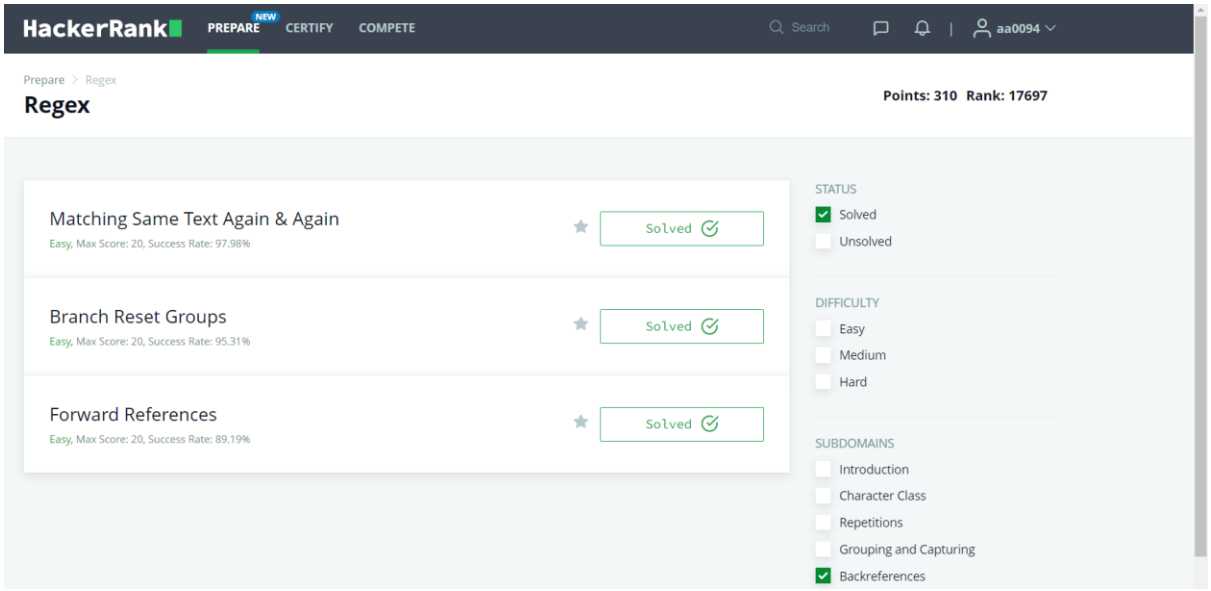


Q.15.Branch reset group



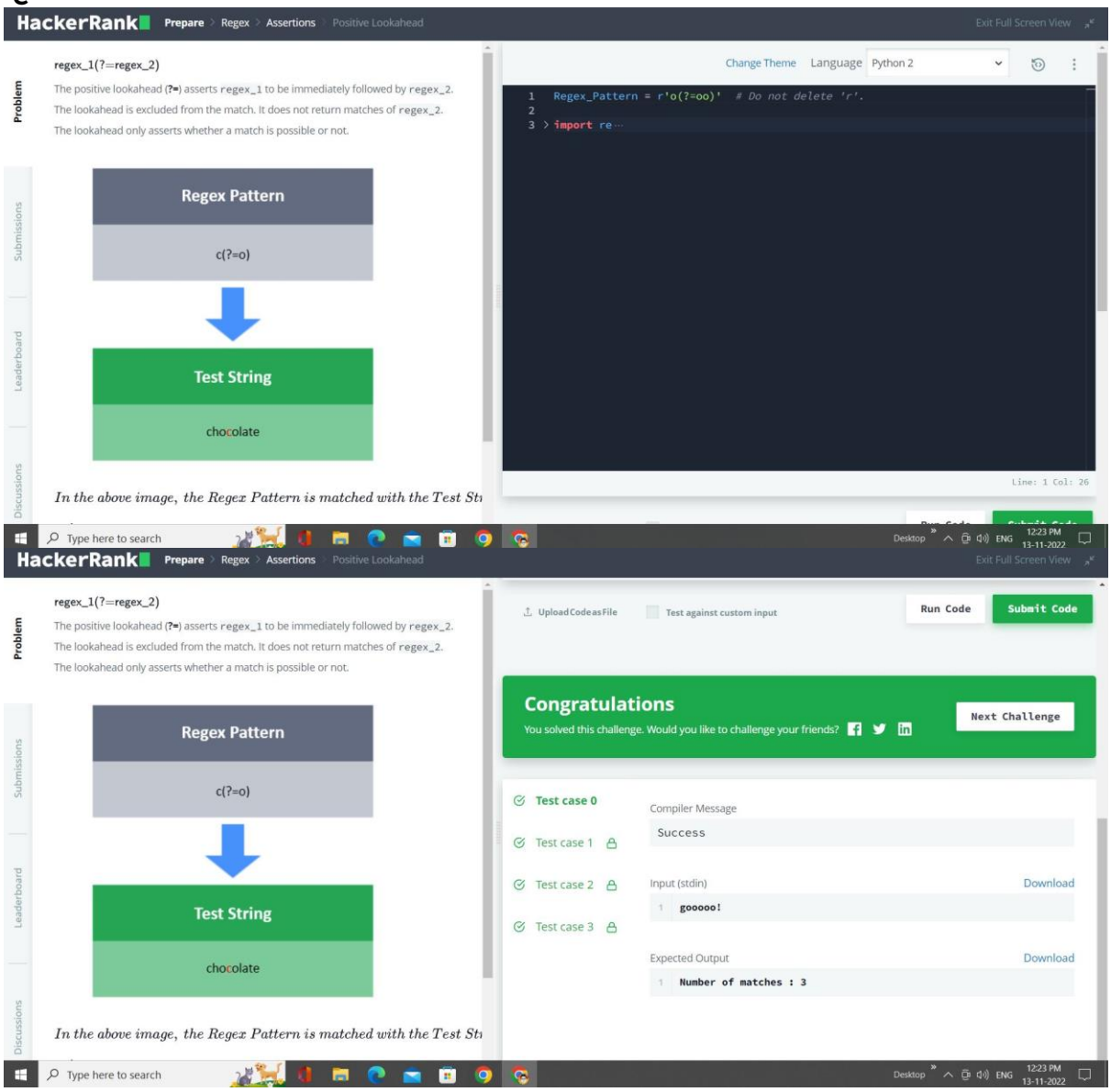
Q.16.Forward References



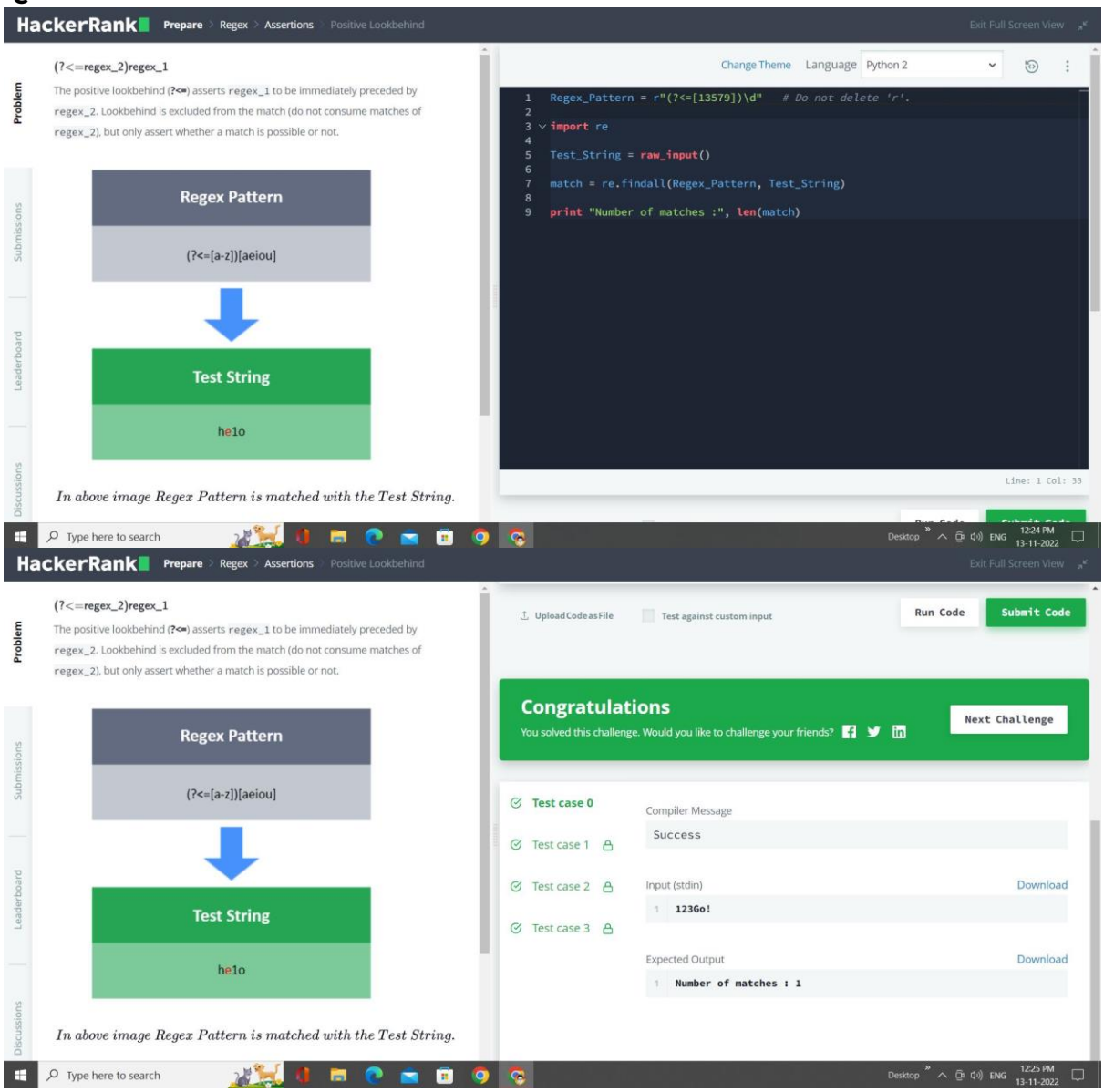

HackerRank
PREPARE
NEW
CERTIFY
COMPETE
Search
aa0094
Prepare > Regex
Regex
Points: 310 Rank: 17697
Matching Same Text Again & Again
Easy, Max Score: 20, Success Rate: 97.98%
Solved
Branch Reset Groups
Easy, Max Score: 20, Success Rate: 95.31%
Solved
Forward References
Easy, Max Score: 20, Success Rate: 89.19%
Solved
STATUS
Solved
Unsolved
DIFFICULTY
Easy
Medium
Hard
SUBDOMAINS
Introduction
Character Class
Repetitions
Grouping and Capturing
Backreferences

Assertions

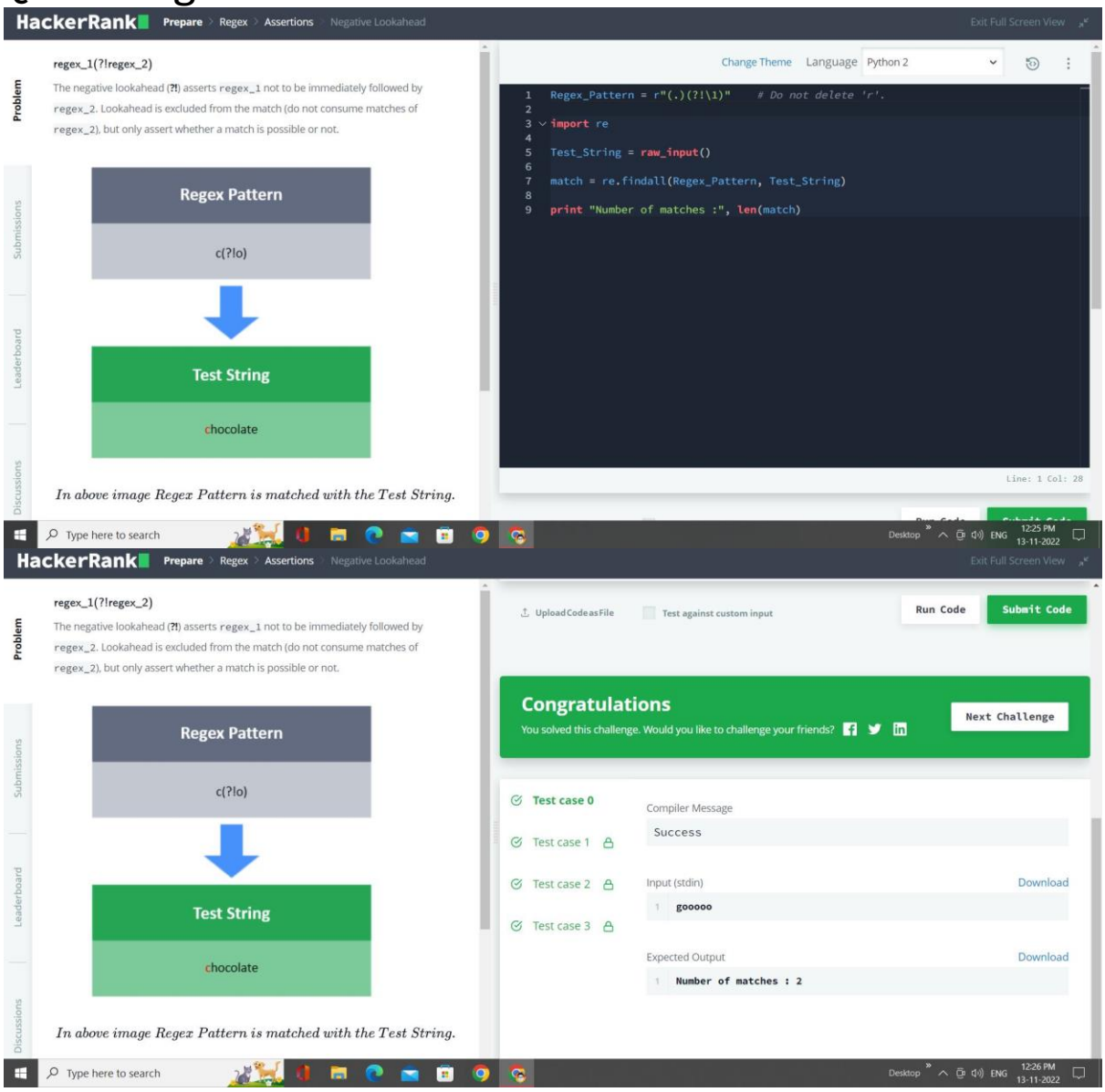
Q.17. Positive Lockahead



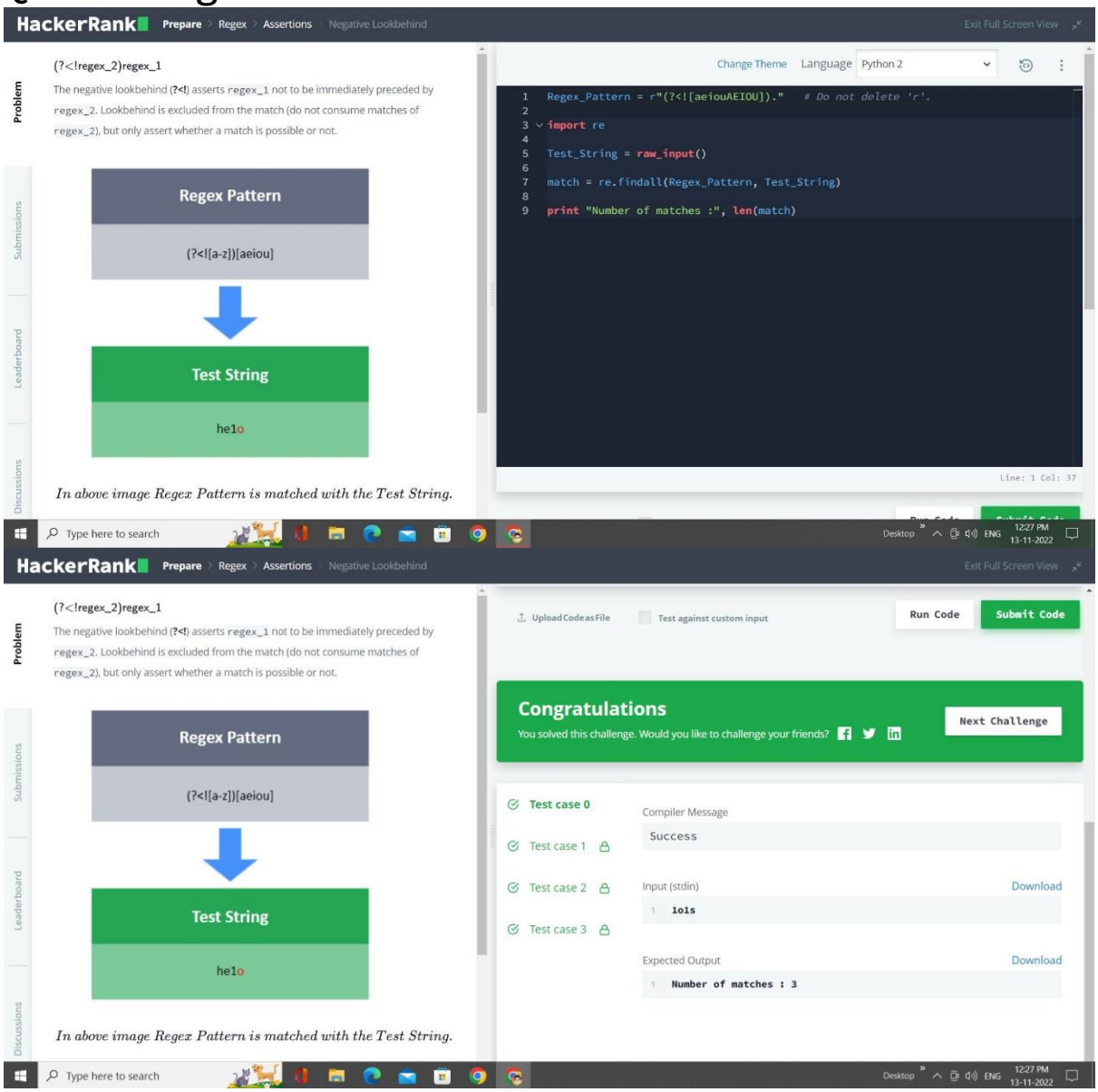
Q.18. Positive Lockbehind

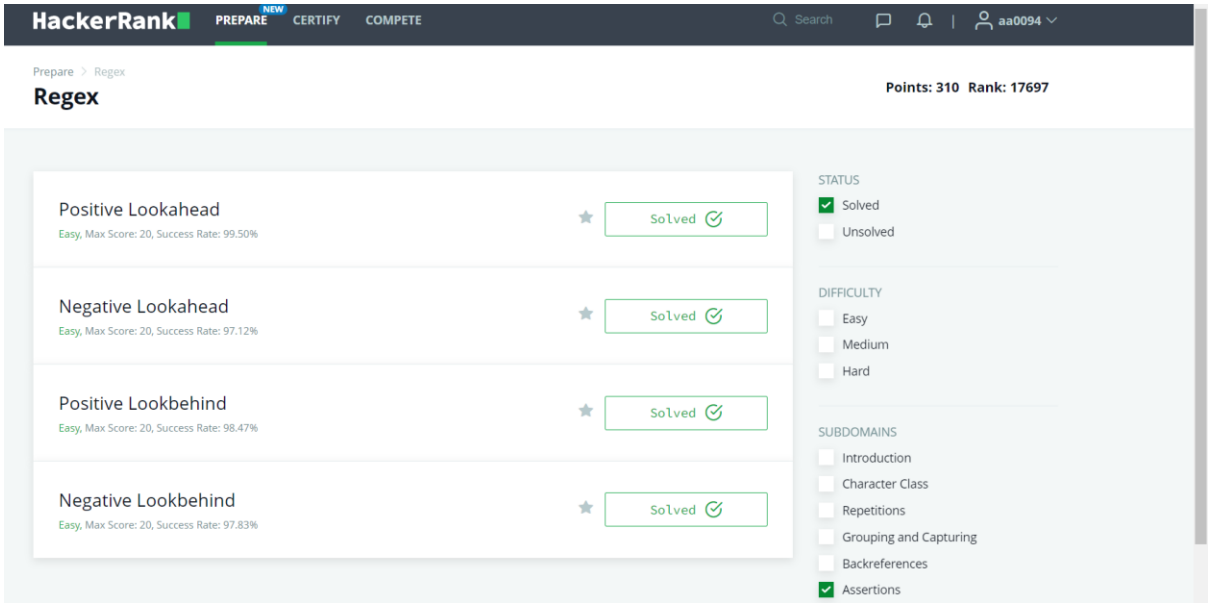


Q.19.Negative Lockahead



Q.20. Negative Lockbehind




HackerRank
PREPARE
NEW
CERTIFY
COMPETE
Search
aa0094
Prepare > Regex
Regex
Points: 310 Rank: 17697
Positive Lookahead
Easy, Max Score: 20, Success Rate: 99.50%
Solved
Negative Lookahead
Easy, Max Score: 20, Success Rate: 97.12%
Solved
Positive Lookbehind
Easy, Max Score: 20, Success Rate: 98.47%
Solved
Negative Lookbehind
Easy, Max Score: 20, Success Rate: 97.83%
Solved
STATUS
Solved
Unsolved
DIFFICULTY
Easy
Medium
Hard
SUBDOMAINS
Introduction
Character Class
Repetitions
Grouping and Capturing
Backreferences
Assertions

Chapterwise Worksheet

Unit 1

UNIT-1.
Worksheet 1.

RA2011031010006
Anas Ahmed Ather
OL CSE-IT.

MCQ.

(1) (b) $P(k) = m(k) + 5$

(2) (c) Trivial proof

3.] ~~None of the above~~ (d) $m^3 + 3m$

4.] (a)

Descriptive Question

1.] $f(n) = (2^{2n} - 1) \,\%\, 3 = 0$

$f(1) \rightarrow (2^2 - 1) \,\%\, 3 = 3 \,\%\, 3 = 0$

$\therefore f(1)$ is true

Assume, $P(n)$ is true for some k, then

$P(k) = 2^{2k} - 1 = 3a$

$\rightarrow P(k+1) \cdot 2^{2k+2} - 1 = 2^{2k} \cdot 4 - 1 = 2^{2k} \cdot 3 + (2^{2k} - 1)$

$= 2^{2k} \cdot 3 + 3a$

$= 3(2^{2k} + a)$

and $3(2^{2k} + a)$ is divisible by 3

$\therefore$ by mathematical induction statement is proved.

2.] let $a^2 = 3k$ — (1)

and $a = 3k + 1$ — (2)

$a = 3k + 2$ — (3)

On squaring (2) $\rightarrow$ $a^2 = 9k^2 + 1 + 3k$ — (4)

On squaring (3) $\rightarrow$ $a^2 = 9k^2 + 4 + 6k$ — (5)

Since (4) and (5) are not divisible by 3, it is proved by contradiction.

3.) a is odd $= 2n+1$
b is even $= 2n$

$a+b = 4n+1$ which is always odd.
(proved)

4) $P = 2n^2 + 31 - 16n$
by counter example
$n = 5$
$P \Rightarrow 2(5)^2 + 31 - 16(5)$
$\Rightarrow P = 1$ (positive)

$n = 4$
$P \Rightarrow 2(4)^2 + 31 - 16(4)$
$\rightarrow P = -1$ (negative)

$\therefore$ P is not always true

(5) $n >= 5$, $2n > n^2$

$n >= 5 \longrightarrow 2n < n^2$

$n = 5 \longrightarrow 10 < 25$

for $n = k \rightarrow 2k < k^2$

for $n = k+1 \rightarrow 2k+2 < (k+1)^2$

$\Rightarrow 2k+2 < k^2 + 1 + 2k$

$= 2k < k^2 + 2k - 1$

and $k^2 + 2k - 1 > k^2$, for $k \geq 5$

$\therefore$ from induction it's proved.

6.) $P(n) = 1^2 + 2^2 + 3^2$

$$n^2 = \frac{n(n+1)(2n+1)}{6}$$

for $n=1$

$$P(n) = \frac{1 \times 2 \times 3}{6} = 1 \text{ (true)}$$

for $n=k \rightarrow P(n) = \frac{1k \times (1k+1)(2k+1)}{6}$

for $n=k+1 \rightarrow$

$$P(k) = 1^2 + 2^2 + 3^2 + \cdots + k^2 + (k+1)^2$$

$$= \frac{k(k+1)(2k+1)}{6} + (k+1)^2$$

$$= \frac{(k+1)\,[k(2k+1) + (6k+1)]}{6}$$

$$= \frac{(k+1)(k+2)+(2k+3)}{6}$$

$\therefore$ It is true for $P(k+1)$

Unit 2

CT1 - Worksheet - 2 — RA2011031010006 — Anay Ahmed Ather — 01 — CSE-IT

Part - A.

1.] (i) Set of all strings starting and ending with 1's with atleast '10'.

2.] (ii) $R = (0+1)^* 1001 (0+1)^*$

3.] (c) (ii) & (iii)

4.] (i) Regular Languages.

5.] (c) The set of all strings containing at least two 0's.

Part - B.

1.] The language generated by RE $0^*(101)^*11$ is the string begening with 0 & ending with 11 and containing 101 ..... 1 = {0101111}

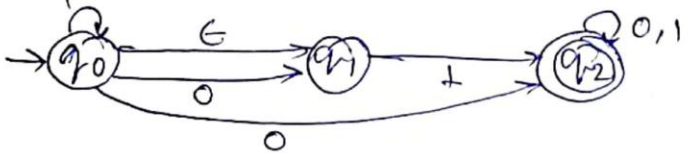
2.] RE is $(01)^*(0+1)^*01$

3.] Yes, it is possible to create such a scenario take any substring of the alphabet set and use $(String)^+$ where + signifies a repetition of for more than 1 times.

Eg: $(a+b)^*(ab)^+(a+b)$

4.] ϵ closure of stat q is all the following transition out of q that are pabelled

ϵ closure of $q_0 \rightarrow \{q_0, q_1, q_2\}$



ϵ closure of q_0

$\rightarrow \epsilon^* \quad 0 \quad \epsilon^*$ $\qquad$ $\epsilon^* \perp \epsilon^*$

$q_0 \rightarrow q_1 \rightarrow \phi \rightarrow \phi$ $\qquad$ $q_0 \rightarrow q_1 \Rightarrow q_1 \rightarrow q_1$,

$\therefore$ ϵ^* of $q_0 \Rightarrow \{q_0, q_1\}$.

8:] 5 tuple structure of NFA & DFA

DFA $\rightarrow$ $\{Q, \Sigma, q_0, F, \delta\}$

$Q \rightarrow$ set of all set

$\Sigma \rightarrow$ Input symbols

$q_0 \rightarrow$ Initial state

$F \rightarrow$ final state

$\delta \rightarrow$ Transition fn $\rightarrow Q \times \Sigma = Q$

NFA $\rightarrow$ $\{Q, \Sigma, q_0, F, \delta\}$

$Q \rightarrow$ set of all states

$\Sigma \rightarrow$ Input symbols.

$q_0 \rightarrow$ Initial states

$F \rightarrow$ final states

$\delta \rightarrow$ Transition fn $\rightarrow Q \times F = 2^Q$

Unit 3

Part - A.

01. CSE-IT

1.] (i) Set of all strings starting and ending of 1's with a two '10'.

2.] (ii) $n = (0+1)^* 1001 (0+1)^*$

3.] (c) (ii) & (iii)

4.] (i) Regular Languages.

5.] (c) The set of all strings containing at least two 0's.

Part - B.

1.] The language generated by RE $0^*(101)^*11$ is the string begening with 0 & ending with 11 and containing 101 ..... 1 = {0101111}

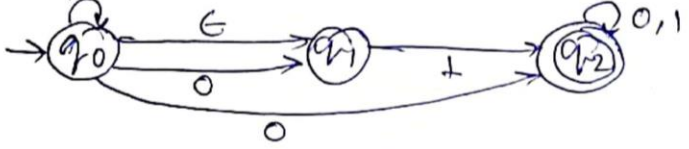
2.] RE is $(01)^*(0+1)^* 01$

3.] Yes, it is possible to create such a scenario take any subsitring of the alphabet set and use $(String)^+$ where + signifies a repetition of for more than 1 times.

Eg: $(a+b)^* (ab)^+ (a+b)$

4.] $\in$ closure of state q is all the following transition out of q that are labelled

$\in$ closure of $q_0 \rightarrow \{q_0, q_1, q_2\}$.



Unit 4

Unit-4 (Worksheet)-1

Anas Ahmed Ather.

1) True

2) True

3) (b) 7

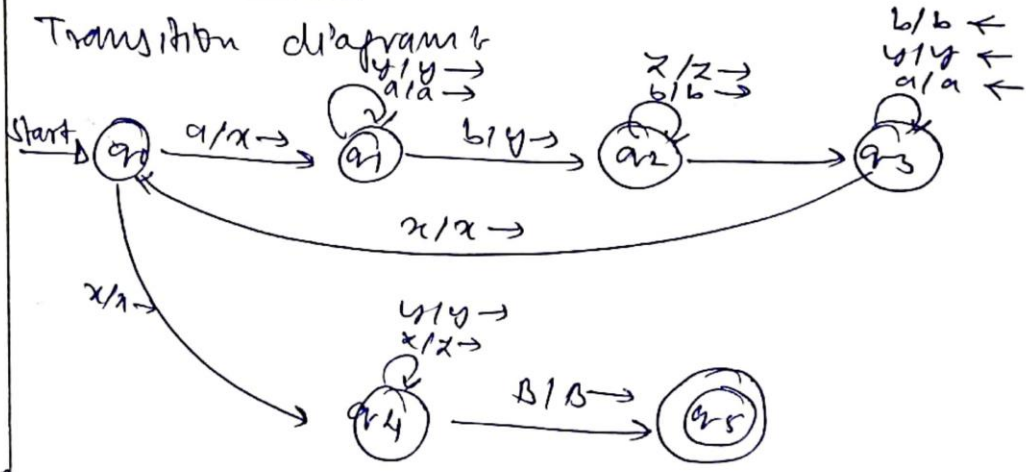
4) A. Left

5) A. Both S1 and S2 are true.

6)

Scenario based

1.) Transition diagram :



3) TM M is

$M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0,1\}, \{0, 1, X, Y, B\}, \delta, q_0, B, \{q_4\})$

where δ is given.

State	0	1	x	y	B
q_0	(q_1, X, R)	–	–	(q_3, Y, R)	–
q_1	$(q_1, 0, R)$	(q_2, Y, L)	–	(q_1, Y, R)	–
q_2	$(q_2, 0, L)$	–	(q_0, X, R)	(q_2, Y, L)	–
q_3	–	–	–	(q_3, Y, R)	(q_4, B, R)
q_4	–	–	–	–	–

Unit 5

UNIT-5 (Worksheet.)

1.) Assume H = Head
T = Tail

Answer Attached Above

LIST R = (T, H, HTH, TT)

LIST S = (TH, TH, HT, T)

Now, we have to find out a sequence that strings formed by R and S are identical

such a sequence is 1, 2, 1, 3, 3, 4.

Hence from the R and S list

1	2	1	3	3	4	1	2	1	3	3	4
T	H	T	HTH	HTH	TT	TH	TH	TH	HT	HT	T

(OR)

i	LIST R w_i	LIST S x_i
1	T	TH
2	H	TH
3	HTH	HT
4	TT	T

Take M = 5

Take the combination 1 2 1 3 3 4

THTHTH HTHTT = TH THTHHTHTT

Distance of PCP = 1 2 1 3 3 4

4) In Excel, there -

Sol) Counting sort takes $O(n+k)$ time and $O(n+k)$ space, where n is the number of items we're sorting and k is the no. of possible values.

We iterate through the input items twice - once to populate counts and once to fill in the output array. Both iterate are $O(n)$ time. Additionally, we iterate through counts once to fill in next index, which is $O(k)$ time.

The algorithm allocates three additional arrays one for count one for next index, and one for the output. The first two are $O(k)$ space and the final one is $O(n)$ space.

[redacted] [redacted]

Note: Enclose the assignment and relevant certificates along with the profile