

STUDENT PORTFOLIO



Name: JAYANT CHOUBEY
R. No.: RA1911003010277
Mail ID: jc8047@srmist.edu.in
Department: B.Tech
Specialization: CSE (core)
Semester: 5

Subject Title: 18CSE301T- Formal Language Automata

Handled By: Uma Devi M

ASSIGNMENT

Assignment questions were both easy and hard and i solved them with my understanding of concepts and notes provided by the teacher.

ASSIGNMENT

The subject was interesting and I enjoyed solving the questions.



18CSC301T – FORMAL LANGUAGE AND AUTOMATA

“Assignment - 1”

Name: JAYANT CHOUBEY
Course: B. Tech Dept.: CSE Sec.: E1
Registration No.: RA1911003010277

FLA Assignment - 1

JAYANT CHOUBEY
RA1911003010277
E1

- ① Finite automata lacks memory or have no auxiliary storage. It remembers something by changing a state. So we may say it has finite storage capability. Memory in finite automata is present in the form of states & only and according to automata principle any automata can have only finite set of states.

e.g. finite automata has finite memory, this is the reason automata for regular language caused finite automata.

We can say that finite automata has finite control, as it decides the next state on receiving particular i/p for i/p tape.

The tape reader reads the cells one by one from L \rightarrow R and at a time only one i/p symbol is read.

- ② $A = \{1^{n^2}, n \geq 0\}$ ~~is not~~

using defining finite automata :-

~~will use~~

Now $A = \{1^{n^2}, n \geq 0\}$

$n \rightarrow$ belongs to \mathbb{N} (natural numbers)
which is infinite.

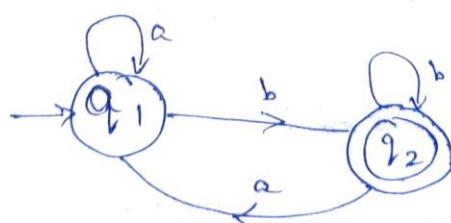
Now infinite language is not regular. In some cases they are regular. That case is when we generate strings with some AP, and we find such patterns in it, so we can make DFA ~~for~~ which becomes regular.

Now $n = 1, 2, 3, 4$

$$L = \{a^0, a^1, a^2, a^3, \dots\}$$

This is infinite and does not produce AP and we can't find such pattern in it, \therefore we can make DFA from it.
 \Rightarrow this language is not regular.

③ DFA $\rightarrow L = \{a, b\}$



$$R_{ijk} = R_{ijk^1} + R_{ijk^1} [R_{kk^{k-1}}]^* \cdot R_{kj^{k^1}}$$

$i=1$

$j=2$

$k=2$

$$R_{12}^2 = R_{12} + R_{12} [R_{22}]^* \cdot R_{22}$$

Now '0' immediate states are :-

$$R_{11}^0 = \underline{a + \epsilon} = a$$

$$R_{21} = a$$

$$R_{12}^0 = b$$

$$R_{22} = \underline{\epsilon + b} = b$$

Now,

$$\begin{aligned} R_{12} &= R_{12}^0 + R_{11}^0 [R_{11}^0]^* \cdot R_{12}^0 \\ &= b + a(a)^* b \\ &= \cancel{a} b + \cancel{a}^* b \\ &= \cancel{a} b a^* b \end{aligned}$$

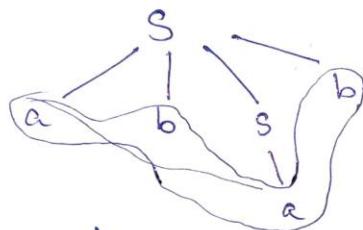
$$\begin{aligned} R_{22} &= R_{22}^0 + R_{21}^0 [R_{11}^0]^* \cdot R_{12}^0 \\ &= b + a [a^*] b \\ &= b + a^* b \\ &= a^* b \end{aligned}$$

$$\begin{aligned}
 R_{12^2} &= R_{12'} + R_{12'} (R_{22'})^* R_{22'} \\
 &= a^* b + a^* b (a^* b)^* . a^* b \\
 &= \cancel{(a^* b)^*} \cdot a^* b \cancel{(a^* b)^*}
 \end{aligned}$$

(ii) (a) $w = 'abab'$

$$S \Rightarrow a | aAb | absb$$

$$A \Rightarrow a \wedge Ab | bs$$

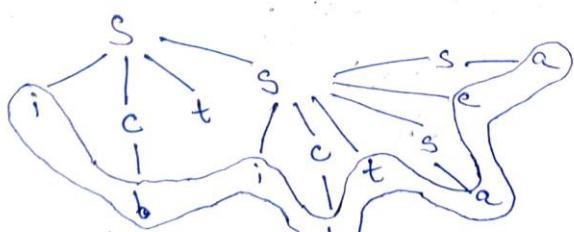


Now trees can be constructed for the given grammar, it is ambiguous.

$$(b) S \Rightarrow ictseclists | a$$

$$c \Rightarrow b$$

$$w = ibtibt aea$$



Here string can be derived using 2 trees
Hence the given grammar is ambiguous.



SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

18CSC301T – FORMAL LANGUAGE AND AUTOMATA

“Assignment - 2”

Name: JAYANT CHOUBEY

Course: B. Tech Dept.: CSE Sec.: E1

Registration No.: RA1911003010277

FLA Assignment - 2

JAYANT CHOUBEY
RA1911003010277
E1

③ $S \rightarrow ASA/aB/b$

$$A \rightarrow B$$

$$B \rightarrow \epsilon/b$$

There are 2 nullable variables - A and B

At first, we will remove $B \rightarrow \epsilon$

After removing $B \rightarrow \epsilon$, the production set

becomes - $S \rightarrow ASA/aB/b/a, A\epsilon B/b/a,$

$$A\epsilon B/b \text{ & } \epsilon$$

$$B \rightarrow b$$

Now we will remove $A \rightarrow \epsilon$

After removing $A \rightarrow \epsilon$, the production set becomes :-

$$S \rightarrow ASA/aB/b/a/SA/As/S$$

$$A \rightarrow B/b$$

$$B \rightarrow b$$

This is the final production set without null transition.

(6)

$$(a) S \rightarrow aB \mid bA$$

$$A \rightarrow a \mid s \mid bAA$$

$$B \rightarrow b \mid bS \mid aBB$$

$$S \rightarrow aB \text{ (rule 1)}$$

$$\rightarrow aabB \text{ (rule 3)}$$

$$\rightarrow aabbB \text{ (rule 3)}$$

$$\rightarrow aabbbS \text{ (rule 3)}$$

$$\rightarrow aabbbaB \text{ (rule 1)}$$

$$\rightarrow aabbabS \text{ (rule 3)}$$

$$\rightarrow aaabba bba \text{ (rule 1)}$$

$$\rightarrow aaabbabbba \text{ (rule 2)}$$

(b)

$$S \rightarrow OB \mid IA$$

$$A \rightarrow O \mid OS \mid IAA$$

$$B \rightarrow I \mid IS \mid OBB$$

$$S \rightarrow OB \text{ (rule 1)}$$

$$\rightarrow OIS \text{ (rule 3)}$$

$$\rightarrow OIIA \text{ (rule 1)}$$

$$\rightarrow OII O \text{ (rule 2)}$$

$$\textcircled{7} \quad S \rightarrow aCa$$

$$C \rightarrow aCa/b$$

$$S \rightarrow aCa$$

$$\rightarrow aaCaa$$

$$\vdots$$

$$\rightarrow a^n Ca^n$$

$$\rightarrow a^n ba^n$$

In the ~~Del~~ language generated by the grammar, there will be atleast one C. Null does not belong to the language set.

\therefore the language generated by the grammar

$$L(G) = a^n b a^n, n > 0$$

\textcircled{8}

(a)

$$S \rightarrow A sB / \epsilon$$

$$A \rightarrow aAS/a$$

$$B \rightarrow S_b S/a/bb$$

Step 1 \rightarrow as start symbol S appears on RHS, we will create a new production rule $S_0 \rightarrow S$.

Therefore, the grammar will become:-

$$S_0 \rightarrow S$$

$$S \rightarrow A sB / \epsilon$$

$$A \rightarrow aAS/a$$

$$B \rightarrow S_b S/a/bb$$

eliminating ϵ -productions

$$S \rightarrow ASB | AB$$

$$A \rightarrow aAS | a | aA$$

$$B \rightarrow SbS | A | bb | bS | Sb | b$$

eliminating any unit productions

$$S \rightarrow ASB | AB$$

$$A \rightarrow aAS | a | aA$$

$$B \rightarrow SbS | aAS | a | aA | bb | bS | Sb | b$$

Introducing a variable and productions

$$C \rightarrow a \text{ and } D \rightarrow b,$$

$$S \rightarrow ASB | AB$$

$$A \rightarrow CA S | a | cA$$

$$B \rightarrow SDS | CA S | a | cA | DD | DS | SD | b$$

$$C \rightarrow a$$

$$D \rightarrow b$$

CNF grammar is:-

$$S \rightarrow AE | AB$$

$$A \rightarrow CF | a | cA$$

$$B \rightarrow SC | FC | a | cA | DD | DS | SD | b$$

$$C \rightarrow a$$

$$D \rightarrow b$$

$$E \rightarrow SB$$

$$F \rightarrow AS$$

$$G \rightarrow DS$$

(9)
(10)

$$S \rightarrow axbx$$

$$x \rightarrow ay/by/\epsilon$$

$$y \rightarrow x/c$$

$$z \rightarrow ax/y$$

elimination of ϵ :

$$S \rightarrow axbx/abx/axb/ab$$

$$x \rightarrow ay/by/a/b$$

$$y \rightarrow x/c$$

$$z \rightarrow ax/y$$

$$y \rightarrow x$$

$$\Rightarrow \cancel{S} \rightarrow axbx/abx/axb/ab$$

$$x \rightarrow ay/by/a/b$$

$$y \rightarrow ay/by/a/b/c$$

$$z \rightarrow ax/y$$



$$z \rightarrow y$$

$$\Rightarrow S \rightarrow axbx/abx/axb/a/b$$

$$x \rightarrow ay/by/a/b$$

$$y \rightarrow ay/by/a/b/c$$

$$z \rightarrow ax/ay/by/a/b/c$$

Now break up the RHS of S ; and replace a by A , b by B and c by C whenever

not units :-

$$S \rightarrow EF | AF | EB | AB$$

$$X \rightarrow AY | BY | a | b$$

$$Y \rightarrow AY | BY | a | b | c$$

$$Z \rightarrow AY | AY | BY | a | b | c$$

$$F \rightarrow AX$$

$$F \rightarrow BX$$

$$A \rightarrow a$$

$$B \rightarrow b$$

~~$$C \rightarrow c$$~~

(b)

$$S \rightarrow aAa | aBb | \epsilon$$

$$A \rightarrow C | a$$

$$B \rightarrow C | b$$

$$C \rightarrow CDE | \epsilon$$

$$D \rightarrow A | B | ab$$

removing E-transitions :-

$$S \rightarrow aAa | bBb | aa | bb$$

$$A \rightarrow C | a$$

$$B \rightarrow C | b$$

$$C \rightarrow CDE | DE$$

$$D \rightarrow A | B | ab$$

removing all unit productions :-

$S \rightarrow aAa \mid bBb \mid aa \mid bb$

$A \rightarrow CDE \mid DE \mid a$

$B \rightarrow CDE \mid DE \mid b$

$C \rightarrow CDE \mid DE$

$D \rightarrow CDE \mid DE \mid ab \mid a \mid b$

eliminate useless symbols

$S \rightarrow aAa \mid bBb \mid aa \mid bb$

$A \rightarrow a$

$B \rightarrow b$

put into CNF

$S \rightarrow [aa]A \mid [bb]B \mid AA \mid BB$

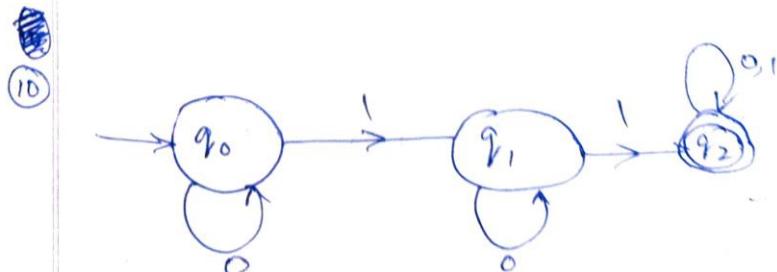
$A \rightarrow a$

$B \rightarrow b$

$[aa] \rightarrow AA$

$[bb] \rightarrow BB$

This gives our grammar in chunky normal form.



$$R_{ij}^k = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} [R_{kk}^{(k-1)}]^* R_{kj}^{(k-1)}$$

$$R_{13}^3 = R_{13}^2 + R_{13}^2 [R_{33}^2]^* R_{33}^2$$

Intermediate state :-

$$R_{11}^{\circ} = \varepsilon + 0 \quad R_{21}^{\circ} = \phi \quad R_{31}^{\circ} = \phi$$

$$R_{12}^{\circ} = 1 \quad R_{22}^{\circ} = \varepsilon + 0 \quad R_{32}^{\circ} = \phi$$

$$R_{13}^{\circ} = \phi \quad R_{23}^{\circ} = 1 \quad R_{33}^{\circ} = \varepsilon + 0 + 1$$

$$R_{13}^1 = R_{13}^{\circ} + R_{12}^{\circ} [R_{22}^{\circ}]^* R_{23}^{\circ}$$

$$R_{33}^2 = R_{33}^{\circ} + R_{32}^{\circ} + [R_{22}^{\circ}]^* R_{23}^{\circ}$$

$$\begin{aligned} R_{13}^1 &= R_{13}^{\circ} + R_{11}^{\circ} [R_{11}^{\circ}]^* R_{13}^{\circ} \\ &= \phi + (\varepsilon + 0) [\varepsilon + 0]^* \phi = \phi \end{aligned}$$

$$\begin{aligned} R_{12}^1 &= R_{12}^{\circ} + R_{11}^{\circ} [R_{11}^{\circ}]^* R_{12}^{\circ} \\ &= 1 + (\varepsilon + 0) (\varepsilon + 0)^* 1 = 1 + 0 0^* 1 \end{aligned}$$

$$\begin{aligned} R_{22}^1 &= R_{22}^{\circ} + R_{21}^{\circ} [R_{11}^{\circ}]^* R_{12}^{\circ} \\ &= (\varepsilon + 0) + \phi = \varepsilon + 0 \end{aligned}$$

$$\begin{aligned} R_{23}^1 &= R_{23}^{\circ} + R_{21}^{\circ} [R_{11}^{\circ}]^* R_{13}^{\circ} \\ &= 1 + \phi = 1 \end{aligned}$$

$$\begin{aligned} R_{33}^1 &= R_{33}^{\circ} + R_{31}^{\circ} [R_{11}^{\circ}]^* R_{13}^{\circ} \\ &= (\varepsilon + 0 + 1) + \phi = \varepsilon + 0 + 1 \end{aligned}$$

$$\begin{aligned} R_{32}^1 &= R_{32}^{\circ} + R_{31}^{\circ} [R_{11}^{\circ}]^* R_{12}^{\circ} \\ &= \phi + \phi = \phi \end{aligned}$$

$$R_{33}^2 = (\varepsilon + 0 + 1) + \phi = \varepsilon + 0 + 1$$

Now

$$R_{13}^3 = \left(\left(1 + 00^* 1 \right) 0^* 1 \right) + \left(\left(1 + 00^* 1 \right) 0^* 1 \right) (\varepsilon + 0 + 1)^* (\varepsilon + 0 + 1)$$
$$- \left(\left(1 + 00^* 1 \right) 0^* 1 \right) + \left(\left(1 + 00^* 1 \right) 0^* 1 \right) \\ (0 + 1)^*$$



18CSC301T – FORMAL LANGUAGE AND AUTOMATA

“NOTES”

Name: JAYANT CHOUBEY
Course: B. Tech Dept.: CSE Sec.: E1
Registration No.: RA1911003010277

Alphabets (Σ)

It's a finite non-empty set of symbols.

Σ is the symbol used.

$\Sigma = \{0, 1\} \rightarrow$ binary alphabets.

$\Sigma = \{a, b, c, \dots\} \rightarrow$ lower case alphabets.

Strings

finite sequence of symbols chosen from some alphabets.

e.g. 10101

Hello

Empty string

- string with 0 occurrences of symbols.
 ϵ → symbol used

Length of the string (n)

No. of symbols in the string.

e.g. |10101| $\rightarrow 5$

|101| $\rightarrow 3$

$|\epsilon| \Rightarrow 0$

Powers of an Alphabet

If Σ is an alphabet, we can express the set of all strings of a certain length from that alphabet by using exponential notation.

e.g. Σ^k - set of strings of length 'k' each of whose symbols is $\in \Sigma$.

$$\textcircled{1} \quad \Sigma^0 = \{\epsilon\}$$

$$\textcircled{2} \quad \text{if } \Sigma = \{0,1\}$$

$$\Sigma^1 = \{0,1\}$$

$$\Sigma^2 = \{00, 01, 10, 11\}$$

and so on.

(3) Σ^* → set of all strings over an alphabet Σ

$$\{0, 1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, \dots\}$$

~~0, 1, 00, 01, 10, 11, ...~~

$$\textcircled{4} \quad \Sigma^* = \{\Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots\}$$

(4) $\Sigma^+ =$ set of non-empty strings from Σ .

$$\Sigma^+ = \{\Sigma^1 \cup \Sigma^2 \cup \dots\}$$

$$\Sigma^* = \Sigma^+ \cup \epsilon$$

$$\Sigma^+ = \Sigma^* - \epsilon$$

Concatenation of strings:

Let x, y be strings, $x \cdot y$ denotes the concatenation of x and y .

Eg. $x = a_1, a_2, a_3, \dots, a_i$

$$y = b_1, b_2, \dots, b_j$$

$$x \cdot y = a_1, a_2, \dots, a_i \cdot b_1, b_2, b_3, \dots, b_j$$

$$|x \cdot y| = i + j$$

$$x = 01101, \quad y = 110$$

$$x \cdot y = 01101110$$

$$y \cdot x = 11001101$$

for any string w ,

$$\epsilon \cdot w = w \cdot \epsilon = w$$

ϵ → identity for concatenation.

Languages:-

A set of strings all of which are chosen from some Σ^* is called language.

If Σ is an alphabet and $L \subseteq \Sigma^*$

then L is a language over Σ .

e.g.

- ① All strings consisting of n 0's followed by n 1's for some $n \geq 0$.

$$L = \{\epsilon, 01, 0011, 000111, \dots\}$$

- ② Set of strings of 0's and 1's with equal no. of each.

$$L = \{\epsilon, 01, 10, 0011, 0101, 1100, \dots\}$$

- ③ Set of binary numbers whose value is a prime

$$L = \{\cancel{0}, 1, 10, 11, 101, 111, \dots\}$$

④ Σ^* is a language for any alphabet Σ .

⑤ \emptyset is the empty language.

⑥ $\{\epsilon\}$ - language consisting of only empty string.

⊗ $\emptyset \neq \{\epsilon\}$

since $\emptyset \rightarrow$ without any string

$\{\epsilon\} \rightarrow$ one empty string.

Proof by induction :-

(1) prove that $1+2+3+\dots+n = \frac{n(n+1)}{2}$

sln:-

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

$n=1$,

$$\text{LHS is } 1, \text{ RHS} = \frac{1(1+1)}{2} = 1$$

LHS=RHS

$P(1)$ is true

when $n=k$, assume $P(k)$ is true

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

~~then~~

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

$$= \frac{(k+1)(k+2)}{2} \quad \text{— RHS}$$

$$\text{LHS} \Rightarrow 1+2+3+\dots+k+k+1$$

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

$$= \frac{(k+1)(k+2)}{2} \quad \text{— LHS}$$

LHS = RHS

Hence proved.

② prove that $11^n - 6$ is divisible by 5 for any integer n .

Sohi: Let $P(n) = \cancel{11^n - 6}$

when $n=1$,

$$P(1) = 11^1 - 6 = 5 \rightarrow \text{divisible by 5}.$$

$P(1)$ is true.

assume $P(k)$ is true, :-

$$P(k) = \cancel{11^k - 6}$$

$$11^k - 6 = 5m \quad \text{for some integer } m.$$

$$\boxed{11^k = 5m + 6}$$

Now, for $P(k+1)$

$$P(k+1) = 11^{k+1} - 6$$

$$\begin{aligned} 11^{k+1} - 6 &= (11 \cdot 11^k) - 6 \\ &= 11(5m + 6) - 6 \\ &= 11(5m) + 66 - 6 \\ &= 55m + 60 \\ &= 5[11m + 12] \end{aligned}$$

It is divisible by 5
 $\Rightarrow P(k+1)$ is true.

Here proved -

Regular expression.

A regular expression is a method used in programming for pattern matching. Regular expressions provide a flexible and concise means to match ~~and~~ strings of text.

Regular expressions can be thought of as the algebraic description of a regular language.

Regular Language : A language is regular if it can be expressed in terms of regular expression.

Rules:-

(1) every letter of the alphabet Σ is a regular expression.

(2) Null string ϵ_0 and empty set \emptyset are regular expressions.

③ If r_1 and r_2 are regular expressions, then

- (i) r_1, r_2
 - (ii) $r_1 r_2$ (concatenation of r_1, r_2)
 - (iii) $r_1 + r_2$ (union of r_1 and r_2)
 - (iv) r_1^*, r_2^* (kleen closure of r_1 and r_2)
- are also regular expressions.

④ \emptyset is a regular expression for ~~the~~ regular language \emptyset .

⑤ ϵ is a regular expression for regular language $\{\epsilon\}$.

⑥ If $a \in \Sigma$, a is a regular expression with language ~~is~~ $\{a\}$.

⑦ If a and b are regular expression, $a+b$ is also a regular expression with language $\{a, b\}$.

~~Order of precedence : Kleen > concatenation~~

<u>Regular expression</u>	<u>Regular set</u>
\emptyset	$\{\emptyset\}$
ϵ	$\{\epsilon\}$
$a+b$	$\{a, b\}$
b^*	$\{\epsilon, b, bb, bbb, \dots\}$
$(a.b^*)$	$\{\epsilon, ab, abb, abbb, \dots\}$
$(0+1)^* 0 (0+1)^* 0 (0+1)^*$	the set of all strings with 2 0's

Identities:-

If A, B, C, D represents regular expressions :-

(a) $\phi^* = \epsilon$

(b) $\epsilon^* = \epsilon$

(c) $AA^* = A^*A$

(d) $(A^*)^* = A^*$

~~(e) $AB = BA$~~

(e) $(AB)^*A = A(BA)^*$

(f) $(a+d)^* = (a^*d^*)^* = (a^* + d^*)^*$

(g) $C + \phi = \phi + C = C$

(h) $A\epsilon = \epsilon A = A$

(i) $\phi B = B\phi = \phi$

(j) $A+A = A$

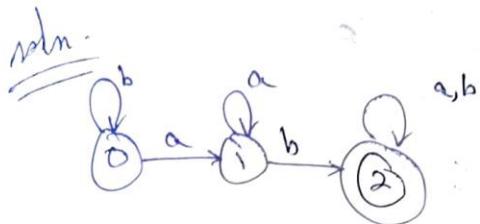
(k) $L(A+B) = LA + LB$

(l) $(A+B)L = AL BL$

(m) $\epsilon + AA^* = \epsilon + A^*A = A^*$

DFA

- (a) $\Sigma = \{a, b\}$
 construct DFA for all strings containing
 ab as substring.



(a) $w = bba\underline{bab}ab$

$$\delta(0, \underline{bb}abab)$$

$$\delta(0, ba\underline{bab})$$

$$\delta(0, ab\underline{ab})$$

$$\delta(1, \underline{bab})$$

$$\delta(2, \underline{ab})$$

$$\delta(2, b)$$

	a	b
0	1	0
1	1	2
2	2	2

2 \rightarrow final state, so accept the string.

(b) $w = aa\underline{a}$

$$\delta(0, \underline{aa}a)$$

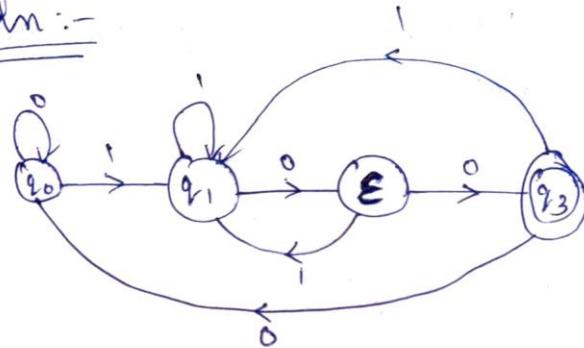
$$\delta(1, aa)$$

$$\delta(1, a)$$

1 \rightarrow not a ~~final~~ final state, so reject
 the string

② $\Sigma = \{0, 1\}$, construct DFA for all strings ending with 100.

solution:-



$w = 011.0100$

$$\delta(q_0, 0110100)$$

$$\delta(q_0, 110100)$$

$$\delta(q_1, 10100)$$

$$\delta(q_1, 0100)$$

$$\delta(q_2, 100)$$

$$\delta(q_1, 00)$$

$$\delta(q_2, 0)$$

$q_3 \rightarrow$ final state

is accept string

	0	1	!
q_0	q_0	q_1	
q_1	q_2	q_1	
q_2	q_3	q_1	
q_3	q_0	q_1	

NFA

NFA accept exactly the regular language
just as DFA's do.

Transition table for NFA:

- (a) each entry in the table for the NFA is a set, even if the set has only one state.
- (b) when there is no transition at all from a gr state on a given slp, the entry in the table is empty.

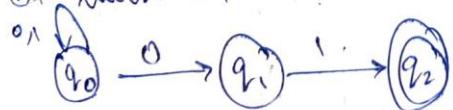
NFA is represented as,

$$A = (\Omega, \Sigma, \sigma, q_0, L)$$

σ is transition fn. that returns a set of states whereas, δ is in DFA.

e.g. NFA that accepts all strings that end

with 01



$$\text{NFA} = (\{q_0, q_1, q_2\}, \{0, 1\}, \sigma, q_0, \{q_2\})$$

from the above NFA,

* σ are set of q_1 , over the slp '0'.

* σ are set of q_2 over the slp '0' & '1'

* no. of arc is 0×2 .

- (*) A DFA has exactly one arc out of each state for each symbol.
 - is such constraint for NFA.

processing of 00101 for the done NFA

step	possible states
0	q_0, q_1
0	q_0, q_1 ↓ ↓ no transition on '0' is q_1 ,
1	q_0, q_2
0	q_0, q_1 ↓ ↓ no transition on '0' is q_2 no transition on 0 in q_1 ,
1	q_0, q_2

→ last state since you are
 in the last symbol.
 q_2 is final state,
 so it accepts the IIP.

	0	1
q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	\emptyset	$\{q_2\}$
$*q_2$	\emptyset	\emptyset

extended transition for σ^*)

Base

$$\sigma^*(q, \epsilon) = \{q\}$$

introduction

Let string of the form $a^n w$

where a - start symbol

w - rest of w

Suppose, $\sigma^*(q, a) = \{p_1, p_2, \dots, p_k\}$.

$$\bigcup_{i=1}^k \sigma^*(p_i, a) = \{r_1, r_2, \dots, r_m\}$$

$$\text{then } \sigma^*(q, w) = \{r_1, r_2, \dots, r_m\}$$

example :-

Let $w = 00101$

$$\textcircled{1} \quad \sigma^*(q_0, \epsilon) = \{q_0\}$$

$$\textcircled{2} \quad \sigma^*(q_0, 0) = \{q_0, q_1\}$$

$$\textcircled{3} \quad \sigma^*(q_0, 00) = \sigma^*(q_0, 0) \cup \sigma^*(q_1, 0)$$

$$= \{q_0, q_1\} \cup \emptyset$$

$$= \{q_0, q_1\}$$

$$\textcircled{4} \quad \sigma^*(q_0, 001) = \sigma^*(q_0, 1) \cup \sigma^*(q_1, 1)$$

$$= \{q_0\} \cup \{q_1\}$$

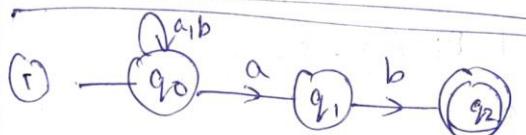
$$= \{q_0, q_1\}$$

$$\begin{aligned}
 ③ \sigma^*(q_0, 0010) &= \sigma(q_{0,0}) \cup \sigma(q_{2,0}) \\
 &= \{q_0, q_1\} \cup \emptyset \\
 &= \{q_0, q_1\}
 \end{aligned}$$

$$\begin{aligned}
 ④ \sigma^*(q_0, 00101) &= \sigma(q_{0,1}) \cup \sigma(q_{1,1}) \\
 &= \{q_0\} \cup \{q_2\}
 \end{aligned}$$

~~q_{1,1}~~

NFA to DFA conversion



$$N = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, q_2)$$

δ -transformation for NFA

	a	b
q ₀	{q ₀ , q ₁ }	{q ₀ }
q ₁	\emptyset	{q ₂ }
q ₂	\emptyset	\emptyset

DFA states.

$$① Q' = \{q_0\}$$

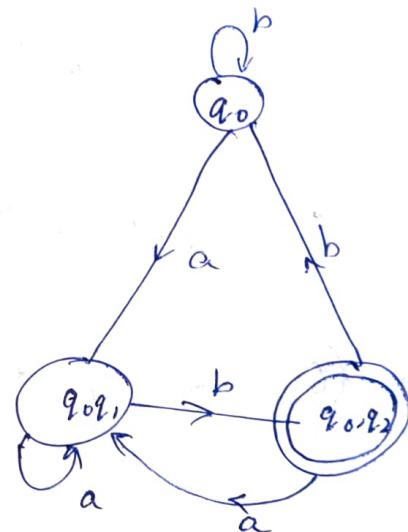
② Now, {q₀, q₁} will be considered as a single state;

$$Q' = \{q_0, \{q_0, q_1\}\}$$

$$\textcircled{3} \quad \sigma'(\{q_0, q_1\}, a) = \sigma'(q_0, a) \cup \sigma(q_1, a) \\ = \{q_0, q_1\}$$

$$\sigma'(\{q_0, q_1\}, b) = \sigma'(q_0, b) \cup \sigma'(q_1, b) \\ = \{q_0, q_2\}$$

state	a	b
q_0	$\{q_0, q_1\}$	q_0
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$



$\textcircled{3}$ $\{q_0, q_2\}$ will be considered as a single state.

$$Q' = \{q_0, \{q_0, q_1\}, \{q_0, q_2\}\}$$

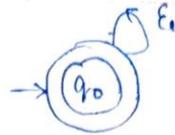
$$\sigma'(\{q_0, q_2\}, a) = \sigma'(q_0, a) \cup \sigma'(q_2, a) \\ = \{q_0, q_1\}$$

$$\sigma'(\{q_0, q_2\}, b) = \sigma'(q_0, b) \cup \sigma'(q_2, b) \\ = \{q_0\}$$

Finite automata for RF.

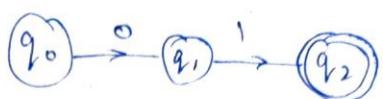
①

$$Y = \epsilon$$



②

$$Y = 01$$



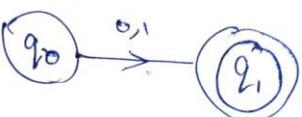
③

$$Y = 011$$



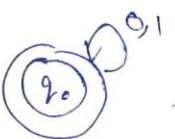
④

$$Y = (0+1)$$



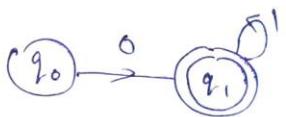
⑤

$$Y = (0+1)^*$$



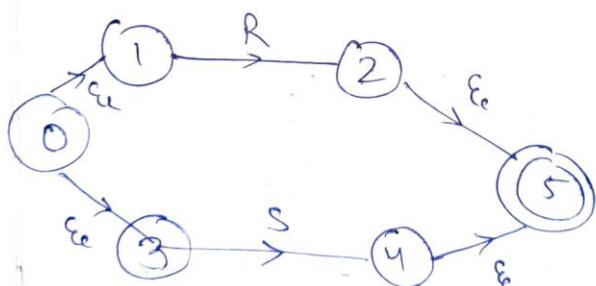
⑥

$$Y = 01^*$$

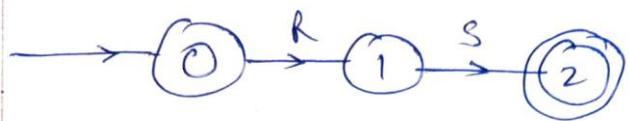


⑦

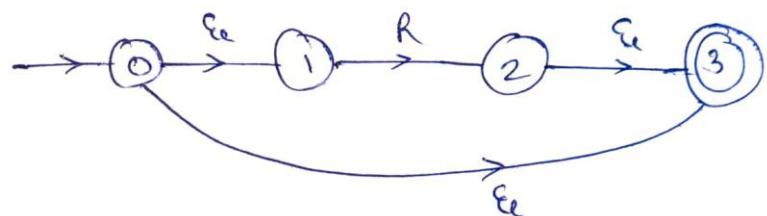
$$R+S$$



② $R \cdot S$



③ closure (R^*)





18CSC301T – FORMAL LANGUAGE AND AUTOMATA

“HACKERRANK ACHIEVEMENTS”

Name: JAYANT CHOUBEY
Course: B. Tech Dept.: CSE Sec.: E1
Registration No.: RA1911003010277

FLA REGEX CODES

CODES:

- 1) DETECT HTML LINKS
- 2) DETECT HTML TAGS
- 3) FIND A SUB-WORD
- 4) ALIEN USERNAME
- 5) IP ADDRESS VALIDATION
- 6) FIND A WORD
- 7) DETECT THE EMAIL ADDRESS
- 8) DETECT THE DOMAIN NAME
- 9) BUILDING A SMART IDE: PROGRAMMING LANGUAGE DETECTION
- 10) SPLIT THE PHONE NUMBER
- 11) HACKERRANK LANGUAGE
- 12) HACKERRANK TWEETS
- 13) BUILD A STOCK EXCHANGE SCRAPER
- 14) UTOPIAN IDENTIFICATION NUMBER
- 15) SAYING HI

Detected HTML links | HackerRank

hackerrank.com/challenges/detect-html-links/problem

HackerRank PREPARE NEW CERTIFY COMPETE APPLY

Search Points: 0 Rank: 104665

Prepare > Regex > Applications > Detect HTML links

Detect HTML links ★

Your Detect HTML links submission got 10.00 points. [Share](#) [Tweet](#)

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Discussions

Charlie has been given an assignment by his Professor to strip the links and the text name from the html pages.

A html link is of the form,

```
<a href="http://www.hackerrank.com">HackerRank</a>
```

Where a is the tag and href is an attribute which holds the link charlie is interested in. The text name is HackerRank.

Charlie notices that the text name can sometimes be hidden within multiple tags.

```
<a href="http://www.hackerrank.com"><h1><b>HackerRank</b></h1></a>
```

Here, the text name is hidden inside the tags h1 and b.

Author: PRASHANTB1984
Difficulty: Medium
Max Score: 10
Submitted By: 9912

NEED HELP?

[View discussions](#) [View top submissions](#)

RATE THIS CHALLENGE

★★★★★

Change Theme Language Python 3

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 if __name__ == '__main__':
5     n = int(input())
6     Regex = r'<a href="(.*?)".*?>([\w ,./]+)(?=)'
7     for _ in range(n):
8         s = input()
9         links = re.findall(Regex, s)
10        for link, att in links:
11            print('%s,%s' % (link, att.strip()))
```

Line: 11 Col: 49

Detect HTML links | HackerRank

Congratulations

You solved this challenge. Would you like to challenge your friends? [Facebook](#) [Twitter](#) [LinkedIn](#)

Next Challenge

Test case 0 Success

Test case 1 Input (stdin) Download

```
2
<p><a href="http://www.quackit.com/html/tutorial/html_links.cfm">Example Link</a></p>
<div class="more-info"><a href="http://www.quackit.com/html/examples/html_links_examples.cfm">More Link Examples...</a></div>
```

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6 Expected Output Download

```
1 http://www.quackit.com/html/tutorial/html_links.cfm,Example Link
2 http://www.quackit.com/html/examples/html_links_examples.cfm,More Link Examples...
```

Detect HTML Tags | HackerRank

HackerRank PREPARE CERTIFY COMPETE APPLY

Search Points: 0 Rank: 104665

Prepare > Regex > Applications > Detect HTML Tags

Detect HTML Tags

Problem Submissions Leaderboard Discussions

In this challenge, we're using regular expressions to detect the various tags used in an HTML document.

- Tags come in pairs. Some tag name, **t**, will have an opening tag, **<t>**, followed by some intermediate text, followed by a closing tag, **</t>**. The forward slash in a closing tag will always come before the tag name.
- The exception to this is self-closing tags, which consist of a single tag (not a pair) with a forward slash after the tag name: **<p/>**

Here are a few examples of tags:

- The **p** tag is for paragraphs: **<p>This is a paragraph</p>**
- There may be 1 or more spaces before or after a tag name: **< p >This is also a paragraph</p>**
- A void or empty tag involves an opening and closing tag with no intermediate characters: **<p></p>**

Some tags can also have attributes, such as the **a** tag, which is used to add a hyperlink to another document:

```
<a href="http://www.google.com">Google</a>
```

In the above case, **a** is the tag name and **href** is an attribute having the value **http://www.google.com**.

Task

Given **N** lines of HTML, find the tag names (ignore any attributes) and print them as a single line of lexicographically ordered semicolon-separated values (e.g.: **tag1;tag2;tag3**).

Input Format

Author PRASHANTB1984

Difficulty Easy

Max Score 10

Submitted By 8675

NEED HELP?

[View discussions](#) [View top submissions](#)

RATE THIS CHALLENGE

MORE DETAILS

[Download problem statement](#) [Download sample test cases](#) [Suggest Edits](#)

[Facebook](#) [Twitter](#) [LinkedIn](#)

The image shows two screenshots of a web browser window for the challenge "Detect HTML Tags" on HackerRank.

Screenshot 1 (Top): A code editor interface. The language is set to Python 3. The code is as follows:

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 if __name__ == '__main__':
5     n = int(input())
6     tags = r'<\s*(\w+)>|<!--[\s\S]*?-->|<script[\s\S]*?>|<style[\s\S]*?>|<[^>]+>'  # Regular expression to find tags
7     ls = set()
8     for _ in range(n):
9         s = input()
10        found = re.findall(tags, s)
11        for i in found:
12            if i not in ls:
13                ls.add(i)
14
15 print(';'.join(sorted(ls)))
```

Screenshot 2 (Bottom): A results page. It says "Congratulations" and "You solved this challenge. Would you like to challenge your friends?". It includes social sharing icons for Facebook, Twitter, and LinkedIn, and a "Next Challenge" button. On the left, it lists "Test case 0" through "Test case 6" with green checkmarks. On the right, it shows the "Compiler Message" (Success), "Input (stdin)" (containing sample HTML code), and "Expected Output" (containing the expected tokens "a;div;p").

Find A Sub-Word | HackerRank

hackerrank.com/challenges/find-substring/problem?h_r=next-challenge&h_v=zen&h_r=next-challenge&h_v=zen

HackerRank PREPARE CERTIFY COMPETE APPLY

Search jc8047

Prepare > Regex > Applications > Find A Sub-Word

Find A Sub-Word ★ Points: 0 Rank: 104665

Your Find A Sub-Word submission got 10.00 points. Share Tweet

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Discussions Editorial

We define a word character to be any of the following:

- An English alphabetic letter (i.e., a-z and A-Z).
- A decimal digit (i.e., 0-9).
- An underscore (i.e., _, which corresponds to ASCII value 95).

We define a word to be a contiguous sequence of one or more word characters that is preceded and succeeded by one or more occurrences of non-word-characters or line terminators. For example, in the string I Love-cheese_?, the words are I, love, and cheese_.

We define a sub-word as follows:

- A sequence of word characters (i.e., English alphabetic letters, digits, and/or underscores) that occur in the same exact order (i.e., as a contiguous sequence) inside another word.

Author dheeraj

Difficulty Easy

Max Score 10

Submitted By 8060

NEED HELP?

View discussions View editorial View top submissions

RATE THIS CHALLENGE

Change Theme Language Python 3

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 if __name__ == '__main__':
5     n = int(input())
6     nline = '\n'.join(input() for _ in range(n))
7
8     q = int(input())
9
10    for _ in range(q):
11        s = input()
12        print(len(re.findall(r'\B(%s)\B' % s, nline)))
```

Line: 12 Col: 55

Upload Code as File Test against custom input Run Code Submit Code

Find A Sub-Word | HackerRank

hackerrank.com/challenges/find-substring/problem?h_r=next-challenge&h_v=zen&h_r=next-challenge&h_v=zen

Upload Code as File Test against custom input Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends? [Facebook](#) [Twitter](#) [LinkedIn](#)

Next Challenge

Test case 0 Compiler Message Success

Test case 1 Input (stdin) Download

```
1 1
2 existing pessimist optimist this is
3 1
4 is
```

Test case 2 Expected Output Download

```
1 3
```

Test case 3

Test case 4

Test case 5

Test case 6

Alien Username | HackerRank

hackerrank.com/challenges/alien-username/problem?h_r=next-challenge&h_v=zen&h_r=next-challenge&h_v=zen&h_r=next-challenge&h_v=zen

HackerRank PREPARE CERTIFY COMPETE APPLY

Search jc8047

Prepare > Regex > Applications > Alien Username Points: 0 Rank: 104665

Alien Username ★

Problem Submissions Leaderboard Discussions Editorial

In a galaxy far, far away, on a planet different from ours, each computer username uses the following format:

1. It must begin with either an underscore, _ (ASCII value 95), or a period, . (ASCII value 46).
2. The first character must be immediately followed by one or more digits in the range 0 through 9.
3. After some number of digits, there must be 0 or more English letters (uppercase and/or lowercase).
4. It may be terminated with an optional _ (ASCII value 95). Note that if it's not terminated with an underscore, then there should be no characters after the sequence of 0 or more English letters.

Given n strings, determine which ones are valid alien usernames. If a string is a valid alien username, print VALID on a new line; otherwise, print INVALID.

Input Format

The first line contains a single integer, n , denoting the number of usernames. Each line i of the n subsequent lines contains a string denoting an alien username to validate.

Constraints

- $1 \leq n \leq 100$

Output Format

Iterate through each of the n strings in order and determine whether or not each string is a valid alien username. If a

Author	dheeraj
Difficulty	Easy
Max Score	10
Submitted By	9675

NEED HELP?

[View discussions](#) [View editorial](#) [View top submissions](#)

RATE THIS CHALLENGE

★ ★ ★ ★ ★

MORE DETAILS

[Download problem statement](#) [Download sample test cases](#) [Suggest Edits](#)

The image shows two screenshots of a web browser window on the HackerRank platform.

Screenshot 1: Code Editor

A code editor window titled "Alien Username | HackerRank". The language is set to Python 3. The code in the editor is:

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 if __name__ == '__main__':
5     n = int(input())
6     r = r'^[_.]d+[a-z|A-Z]*[_]?$'
7     for _ in range(n):
8         s = input()
9         if re.match(r, s):
10             print("VALID")
11         else:
12             print("INVALID")
```

Screenshot 2: Challenge Results

A confirmation message: "Congratulations! You solved this challenge. Would you like to challenge your friends? [Facebook](#) [Twitter](#) [LinkedIn](#)".

Buttons: "Run Code" (disabled), "Submit Code" (green button).

Test cases and compiler message:

- Test case 0:** Compiler Message: Success
- Test case 1:** Input (stdin):
1
3
2 _0898989811abcd_
3 _abce
4 _99090909abcd0
- Test case 2:** Expected Output:
1 VALID
2 INVALID
3 INVALID

IP Address Validation | HackerRank

PREPARE NEW CERTIFY COMPETE APPLY

Search Points: 0 Rank: 104665 jc8047

Prepare > Regex > Applications > IP Address Validation

IP Address Validation ★

Your IP Address Validation submission got 10.00 points. Share Tweet

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Discussions Editorial

You will be provided with N lines of what are possibly IP addresses. You need to detect if the text contained in each of the lines represents an (a)IPv4 address (b)IPv6 address or (c)None of these.

IPv4 was the first publicly used Internet Protocol which used 4 byte addresses which permitted for 2^{32} addresses. The typical format of an IPv4 address is A.B.C.D where A, B, C and D are integers lying between 0 and 255 (both inclusive).

IPv6, with 128 bits was developed to permit the expansion of the address space. To quote from the linked article: The 128 bits of an IPv6 address are represented in 8 groups of 16 bits each. Each group is written as 4 hexadecimal digits and the groups are separated by colons (:). The address 2001:0db8:0000:0000:ff00:0042:8329 is an example of this representation.

Consecutive sections of zeros will be left as they are.

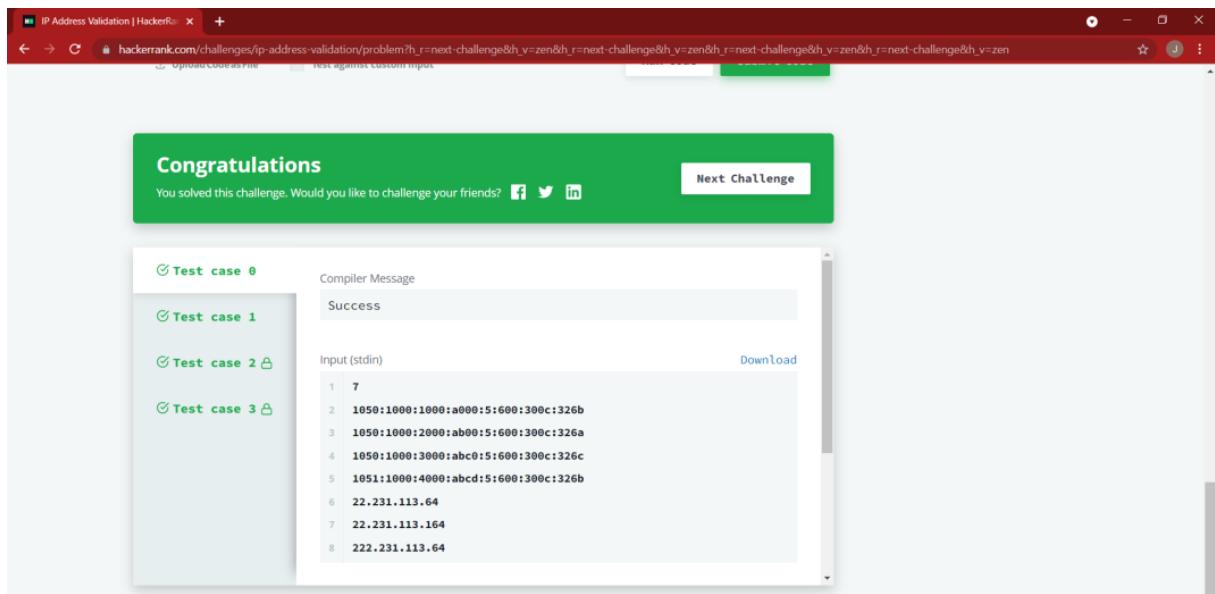
An IPv6 value such as "...:0:..." or "...:5:..." is address-wise identical to "...:0000:..." or "...:0005:...". Leading zeros may be omitted in writing the address.

Author PRASHANTB1984
Difficulty Easy
Max Score 10
Submitted By 10123

NEED HELP?
View discussions View editorial View top submissions

RATE THIS CHALLENGE

```
# Enter your code here. Read input from STDIN. Print output to STDOUT
import re
if __name__ == '__main__':
    n = int(input())
    IPv4_pattern = r'^((\d|[1-9]\d|\d{2}[0-4]\d|\d{2}[0-5])\.)\{3\}(\d|[1-9]\d|\d{2}[0-4]\d|\d{2}[0-5])$'
    for _ in range(n):
        s = input()
        if re.match(IPv4_pattern, s):
            print('IPv4')
        elif re.match(r'^([0-9a-f]{1,4}):{7}[0-9a-f]{1,4}$', s):
            print('IPv6')
        else:
            print('Neither')
```



The screenshot shows the details page for the "Find a Word" challenge. At the top, it says "Find a Word ★ Points: 0 Rank: 104665". Below that, there are tabs for "Problem", "Submissions", "Leaderboard", and "Discussions". The "Problem" tab is selected. It contains the problem statement, which defines a word as a non-empty maximum sequence of characters that can contain only lowercase letters, uppercase letters, digits, and underscores '_'. It also specifies constraints: $1 \leq N \leq 100$ and $1 \leq T \leq 10$. The "Output format" section states that for every word, the number of occurrences of the word in all the N sentences listed should be printed. There are sections for "Constraints", "Input Format", "Output format", and "Sample Input". On the right side, there are details about the author (dheeraj), difficulty (Medium), max score (15), and submission count (7337). There are also links for "View discussions", "View top submissions", "Download problem statement", "Download sample test cases", and "Suggest Edits".

The image shows two screenshots of a web browser displaying a challenge on HackerRank.

Top Screenshot: A code editor window titled "Find a Word | HackerRank". The code is written in Python 3:

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 if __name__ == '__main__':
5     n = int(input())
6     s = [input() for _ in range(n)]
7
8     t = int(input())
9     for _ in range(t):
10        ss = input()
11        Regex = '\\\\b' + ss + '\\\\b'
12        total = 0
13        for i in s:
14            words = re.findall(Regex, i)
15            total += len(words)
16
17 print(total)
```

The status bar at the bottom right indicates "Line: 16 Col: 21". Below the editor are buttons for "Upload Code as File", "Test against custom input", "Run Code", and "Submit Code".

Bottom Screenshot: A results page titled "Congratulations" with a green header. It says "You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#)". There is a "Next Challenge" button.

The results section shows test cases and their outcomes:

- Test case 0:** Success
- Test case 1:** Success
- Test case 2:** Success
- Test case 3:** Success
- Test case 4:** Success
- Test case 5:** Success
- Test case 6:** Success

For each test case, there is an "Input (stdin)" and "Expected Output" section with a "Download" link.

At the bottom of the page, there is a footer with links: Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature.

https://www.hackerrank.com/challenges/detect-the-email-addresses/leaderboard

Detected Email Addresses | Hackerrank

PREPARE NEW CERTIFY COMPETE APPLY

Search jc8047

Prepare > Regex > Applications > Detect the Email Addresses

Detect the Email Addresses ★

Your Detect the Email Addresses submission got 15.00 points.

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Discussions

You will be provided with a block of text, spanning not more than hundred lines. Your task is to find the unique e-mail addresses present in the text. You could use Regular Expressions to simplify your task. And remember that the "@" sign can be used for a variety of purposes! Requirements are simplified versus real world. There can be a number of strings separated by dots before and after the "@" symbol. Strings will be made up of characters in the ranges a-z, A-Z, 0-9, _ (underscore).

Input Format
The first line contains an integer N (N<=100), which is the number of lines present in the text fragment which follows.
From the second line, begins the text fragment (of N lines) in which you need to search for e-mail addresses.

Output Format
All the unique e-mail addresses detected by you, in one line, in lexicographical order, with a semi-colon as the delimiter.

Author PRASHANTB1984
Difficulty Medium
Max Score 15
Submitted By 7823

NEED HELP?
[View discussions](#) [View top submissions](#)

RATE THIS CHALLENGE
★ ★ ★ ★ ★

Change Theme Language Python 3

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 if __name__ == '__main__':
5     n = int(input())
6     ls = set()
7     Regex = r'\b(?:\w|.)\{1,}@(?:\w|\.*\{1,\}\b'
8     for _ in range(n):
9         s = input()
10        found = re.findall(Regex, s)
11        for i in found:
12            if i not in ls:
13                ls.add(i)
14
15 print(';'.join(sorted(ls)))
```

Line: 15 Col: 32

Congratulations
You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#)

Test case 0 Compiler Message
Success

Test case 1

Test case 2

Test case 3

Test case 4

Input (stdin) Download

```
1 36
2 Finally this phone is testimony to our quest and ever open ears
   for hearing from our customers since 1921. We look forward to
   hearing from you today.
3 All India National Toll Free Number: 180 0425 0426
4 Working Hours: 10:00 am to 6:00 pm (Monday ~ Friday),
5 10:00 am to 2:00 pm (Saturday). To report ATM Card Lost, Kindly
   contact: +91 (44) 2622 3106 / 2622 3109.
6 TMB Customer Care: +91 9842 461 461
7 For all your queries, on any of our services in any branch in
   India, you can now SMS @TMBHelpLine, or call +91 9842 461 461
```

Detected the Domain Name ★ Points: 0 Rank: 104665

Your Detect the Domain Name submission got 15.00 points. [Share](#) [Tweet](#)

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Discussions

You will be provided with a chunk of HTML markup. Your task is to identify the unique domain names from the links or URLs which are present in the markup fragment.

For example, if the link <http://www.hackerrank.com/contest> is present in the markup, you should detect the domain: [hackerrank.com](http://www.hackerrank.com). In case there are second level or higher level domains present in the markup, all of them need to be treated as unique. For instance if the links <http://www.xyz.com/news>, <https://abc.xyz.com/jobs>, <http://abcd.xyz.com/jobs2> are present in the markup then [xyz.com, abc.xyz.com, abcd.xyz.com] should all be identified as unique domains present in the markup.

Prefixes like "www." and "ww2.", if present, should be scrubbed out from the domain name.

Input Format:

An Integer N. This is equal to the number of lines in the HTML Fragment which follows. A chunk of HTML Markup with embedded links, the length of which is N lines.

Author: PRASHANTB1984
Difficulty: Medium
Max Score: 15
Submitted By: 5506

NEED HELP?
[View discussions](#) [View top submissions](#)

RATE THIS CHALLENGE

The image shows a browser window with two main sections. The top section is a code editor for a Python challenge titled "Detect the Domain Name". The code uses regular expressions to find URLs in input strings. The bottom section shows the challenge results, indicating success with a green banner and a "Next Challenge" button.

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 if __name__ == '__main__':
5     n = int(input())
6     Regex = r'https://(?:www\.|ww2\.)?([a-zA-Z\d-]+(?:\.[a-zA-Z\d-]+)+)'
7     out = set()
8     for _ in range(n):
9         s = input()
10        temp = re.findall(Regex, s)
11        for i in temp:
12            if i not in out:
13                out.add(i)
14
15 print(';'.join(sorted(out)))
```

Congratulations
You solved this challenge. Would you like to challenge your friends? [Facebook](#) [Twitter](#) [LinkedIn](#) [Next Challenge](#)

Test case 0
Test case 1
Test case 2
Test case 3
Test case 4

Input (stdin)
1 1027
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml" lang="en">
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
6 <meta name="format-detection" content="telephone=no" />
7 <title>Rediff.com - India, Business, Stock, Sports, Cricket,
Entertainment, Bollywood, Music, Video and Breaking news,
On-demand NC Showtime</title>

Compiler Message
Success

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature

The screenshot shows the HackerRank challenge page for 'Building a Smart IDE: Programming Language Detection'. At the top, there's a navigation bar with links for 'PREPARE', 'CERTIFY', 'COMPETE', and 'APPLY'. On the right, it shows the user's points (95) and rank (32847). Below the navigation, the challenge title 'Building a Smart IDE: Programming Language Detection' is displayed with a star icon. A message box at the top center says 'Your Building a Smart IDE: Programming Language Detection submission got 30.00 points.' with sharing options for Facebook and Twitter.

This screenshot shows the detailed view of the challenge. It includes a summary section with instructions about auto-detecting programming languages from source code, sample programs for Java, C, and Python, and a note about using regular expressions. To the right, there's a sidebar with author information (idlecool), difficulty (Medium), max score (30), and submitter count (6235). It also has sections for 'NEED HELP?', 'View discussions', 'View top submissions', and a rating section with five stars.

This screenshot shows the code editor interface where the user can write their solution. The code is written in Python 3 and uses a class named 'Main' to read input from standard input and print output to standard output. It checks if the string 'java' or 'C' is included in the input and prints 'Java' or 'C' respectively. If neither is found, it prints 'Python'. The code editor includes a toolbar with 'Change Theme', 'Language' (set to Python 3), and other icons. The status bar at the bottom indicates 'Line: 19 Col: 17'.

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3 import sys
4
5 class Main:
6     def __init__(self):
7         self.s = ''.join(sys.stdin.readlines())
8
9     def output(self):
10        if 'java' in self.s:
11            print("Java")
12        elif "#include" in self.s:
13            print("C")
14        else:
15            print("Python")
16
17 if __name__ == '__main__':
18     obj = Main()
19     obj.output()
```

Building a Smart IDE: Programming

hackerrank.com/challenges/programming-language-detection/problem

Congratulations

You solved this challenge. Would you like to challenge your friends? [Facebook](#) [Twitter](#) [LinkedIn](#)

[Next Challenge](#)

Test case 0 Success

Test case 1 Success

Test case 2 Success

Test case 3 Success

Test case 4 Success

Test case 5 Success

Test case 6 Success

Compiler Message

Success

Input (stdin)

```
# let us create a test string
testString1 = "Hello World!"
print "Original String: " + testString1
# Print this string in lower case
# Converting a string to lower case
```

Download

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature

Split the Phone Numbers | HackerRank

hackerrank.com/challenges/split-number/problem?h_r=next-challenge&h_v=zen

HackerRank PREPARE CERTIFY COMPETE APPLY

Search jc8047

Prepare > Regex > Applications > Split the Phone Numbers **★** Points: 95 Rank: 32847

Your Split the Phone Numbers submission got 15.00 points. [Share](#) [Tweet](#)

[Try the next challenge](#) | [Try a Random Challenge](#)

Problem Submissions Leaderboard Discussions

There is a list of phone numbers which needs the attention of a text processing expert. As an expert in regular expressions, you are being roped in for the task. A phone number directory can reveal a lot such as country codes and local area codes. The only constraint is that one should know how to process it correctly.

A Phone number is of the following format:

```
[Country code]-[Local Area Code]-[Number]
```

There might either be a '-' (ascii value 45), or a ' ' (space; ascii value 32) between the segments. Where the country and local area codes can have 1-3 numerals each and the number section can have 4-10 numerals each. And so, if we tried to apply the a regular expression with groups on this phone number: 1-425-9854706

Author: dheeraj

Difficulty: Easy

Max Score: 15

Submitted By: 9697

NEED HELP?

[View discussions](#)

[View top submissions](#)

RATE THIS CHALLENGE

★ ★ ★ ★ ★

Split the Phone Numbers | Hackerrank

You can view others' submissions if you solve this challenge. Navigate to the challenge leaderboard.

Change Theme Language Python 3

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 if __name__ == '__main__':
5     n = int(input())
6     Regex = r'(\d*)'
7     for _ in range(n):
8         s = input()
9         a = re.findall(Regex, s)
10        print('CountryCode=%s,LocalAreaCode=%s,Number=%s' %(str(a[0]), str(a[1]), str(a[2])))
```

Congratulations

You solved this challenge. Would you like to challenge your friends? [Facebook](#) [Twitter](#) [LinkedIn](#)

Next Challenge

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Compiler Message

Success

Hidden Test Case

Unlock this testcase for 5 hackos.

Unlock

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature

HackerRank Language | HackerRank

hackerank.com/challenges/hackerrank-language/problem

HackerRank Language ★ Points: 140 Rank: 28442

Your HackerRank Language submission got 15.00 points. [Share](#) [Tweet](#)

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Discussions

Every submission at HackerRank has a field called language which indicates the programming language which a hacker used to code his solution.

C:CPP:JAVA:PYTHON:PERL:PHP:RUBY:CSHARP:HASKELL:CLOJURE:BASH:SCALA:ERLANG:CLISP:LUA:BRAINFUCK:JAVASCRIPT:GO:D:OCAML|R|PASCAL|SBCL|DART|GROOVY|OBJECTIVEC

Sometimes, error-prone API requests can have an invalid language field. Could you find out if a given submission has a valid language field or not?

Input Format

First line contains N. N API requests follow, each in a newline. Each request has an integer api_id and a string language which are the request parameters placed by the an API request.

Constraints

Author dheeraj
Difficulty Easy
Max Score 15
Submitted By 8784

NEED HELP?
[View discussions](#) [View top submissions](#)

RATE THIS CHALLENGE ★ ★ ★ ★ ★

Change Theme Language Python 3

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 if __name__ == '__main__':
5     n = int(input())
6     s = 'C|CPP|JAVA|PYTHON|PERL|PHP|RUBY|CSHARP|HASKELL|CLOJURE|BASH|SCALA|ERLANG|CLISP|LUA|BRAINFUCK|JAVASCRIPT|GO|D|OCAML|R|PASCAL|SBCL|DART|GROOVY|OBJECTIVEC'
7     Regex = r'^[1-9](d{4})\s' + '(' + s + ')$'
8     for _ in range(n):
9         s = input()
10        if re.match(Regex, s):
11            print('VALID')
12        else:
13            print('INVALID')
```

Line: 13 Col: 29

HackerRank Language | HackerRank

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#)

Congratulations

Next Challenge

Test case 0: Success

Test case 1: Success

Test case 2: Success

Test case 3: Success

Test case 4: Success

Test case 5: Success

Test case 6: Success

Compiler Message: Success

Input (stdin):

```
1 3
2 11011 C
3 11022 CPP
4 11044 X
```

Download

Expected Output:

```
1 VALID
2 VALID
3 INVALID
```

Download

HackerRank Tweets | HackerRank

PREPARE NEW CERTIFY COMPETE APPLY

Search jc8047

Prepare > Regex > Applications > HackerRank Tweets

HackerRank Tweets ★ Points: 155 Rank: 27508

Problem Submissions Leaderboard Discussions

Increasing popularity of hackerrank can be seen in tweets like

- I love #hackerrank
- I just scored 27 points in the Picking Cards challenge on #HackerRank
- I just signed up for summer cup @hackerrank

Given a set of most popular tweets, your task is to find out how many of those tweets has the string hackerrank in it.

Input Format

First line is an integer N. N lines follow. Each line is a valid tweet.

Constraints

$1 \leq N \leq 10$

Each character of the tweet is a valid ASCII character.

Output format

Print the total number of tweets that has hackerrank (case insensitive) in it.

Sample Input

```
4
I love #hackerrank
```

Author: dheeraj

Difficulty: Easy

Max Score: 15

Submitted By: 10935

NEED HELP?

[View discussions](#)

[View top submissions](#)

RATE THIS CHALLENGE

★ ★ ★ ★ ★

MORE DETAILS

[Download problem statement](#)

[Download sample test cases](#)

[Suggest Edits](#)

[f](#) [t](#) [in](#)

The image shows a two-panel view of a web application interface for a programming challenge.

Top Panel (Code Editor):

- Header: "HackerRank Tweets | HackerRank" and "hackerank.com/challenges/hackerrank-tweets/problem".
- Toolbar: "Change Theme", "Language" set to "Python 3", and other icons.
- Code Area:

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 if __name__ == '__main__':
5     n = int(input())
6     con = 0
7     for _ in range(n):
8         s = input().upper()
9         if 'HACKERRANK' in s:
10             con += 1
11
12 print(con)
```
- Status: "Line: 12 Col: 15"
- Buttons: "Upload Code as File", "Test against custom input", "Run Code", and a large green "Submit Code" button.

Bottom Panel (Challenge Results):

- Header: "HackerRank Tweets | HackerRank" and "hackerank.com/challenges/hackerrank-tweets/problem".
- Section: "Congratulations" with a message: "You solved this challenge. Would you like to challenge your friends? [Facebook](#) [Twitter](#) [LinkedIn](#)".
- Section: "Compiler Message" showing "Success".
- Section: "Input (stdin)" showing:

```
1 4
2 I love #hackerrank
3 I just scored 27 points in the Picking Cards challenge on
#HackerRank
4 I just signed up for summer cup @hackerrank
5 interesting talk by hari, co-founder of hackerrank
```
- Section: "Expected Output" showing "1 4".
- Buttons: "Next Challenge", "Download" for Input and Output, and "Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature" at the bottom.

Build a Stack Exchange Scraper | +

hackerrank.com/challenges/stack-exchange-scraper/problem

HackerRank PREPARE NEW CERTIFY COMPETE APPLY

Search Points: 170 Rank: 25933

Build a Stack Exchange Scraper ★

Your Build a Stack Exchange Scraper submission got 15.00 points. Share Tweet

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Discussions

Stack Exchange is an information power-house, which contains libraries of crowdsourced problems (with answers) across a large number of topics which are as diverse as electronics, cooking , programming, etc.

We are greatly interested in crawling and scraping as many questions, as we can, from stack-exchange. This is an example of a question library page from stack-exchange.

Your task will be, to scrape the questions from each library page, in the order in which they are listed. You will be provided with the markup of question listing pages, from which you need to detect:

(1) Identifier (2) Question text (which is on the Hyperlink to the question) (3) How long ago the question was asked.

The Markup in the Test Cases will be similar to the sample fragment shown below. Please note, that since this markup is real markup from the website, it is likely to contain some stray control and escape characters, unexpected whitespaces and newlines.

Author PRASHANTB1984

Difficulty Easy

Max Score 15

Submitted By 4307

NEED HELP?

View discussions View top submissions

RATE THIS CHALLENGE

★ ★ ★ ★ ★

MORE DETAILS

Change Theme Language Python 3

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3 import sys
4
5 if __name__ == '__main__':
6     s = sys.stdin.read()
7
8     Regex = r'question-summary-(\w\w\w\w\w\w).+?class="question-hyperlink">(.*?)</a>.+?
9         class="relative-time">(.+?)</span>'
10    li = re.findall(Regex, s, re.DOTALL)
11
12    for a in li:
13        print(''.join(a))
```

Line: 12 Col: 27

Build a Stack Exchange Scraper | +

hackerrank.com/challenges/stack-exchange-scraper/problem

Congratulations

You solved this challenge. Would you like to challenge your friends? [Facebook](#) [Twitter](#) [LinkedIn](#)

[Next Challenge](#)

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Compiler Message

Success

Hidden Test Case

Unlock this testcase for 5 hackos.

[Unlock](#)

[Contest Calendar](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)

Utopian Identification Number | +

hackerrank.com/challenges/utopian-identification-number/problem

HackerRank [PREPARE](#) [NEW](#) [CERTIFY](#) [COMPETE](#) [APPLY](#)

Search [jc8047](#)

Prepare > Regex > Applications > Utopian Identification Number Points: 185 Rank: 24613

Utopian Identification Number ★

Your Utopian Identification Number submission got 15.00 points. [Share](#) [Tweet](#)

[Try the next challenge](#) | [Try a Random Challenge](#)

Problem Submissions Leaderboard Discussions

A new identification number is given for every Citizen of the Country Utopia and it has the following format.

- The string must begin with between 0-3 (inclusive) lowercase letters.
- Immediately following the letters, there must be a sequence of digits (0-9). The length of this segment must be between 2 and 8, both inclusive.
- Immediately following the numbers, there must be at least 3 uppercase letters.

Your task is to find out if a given identification number is valid or not.

Input Format

The first line contains N. N lines follow each line containing an identification number.

Constraints

Author: dheeraj
Difficulty: Easy
Max Score: 15
Submitted By: 7958

NEED HELP?

[View discussions](#) [View top submissions](#)

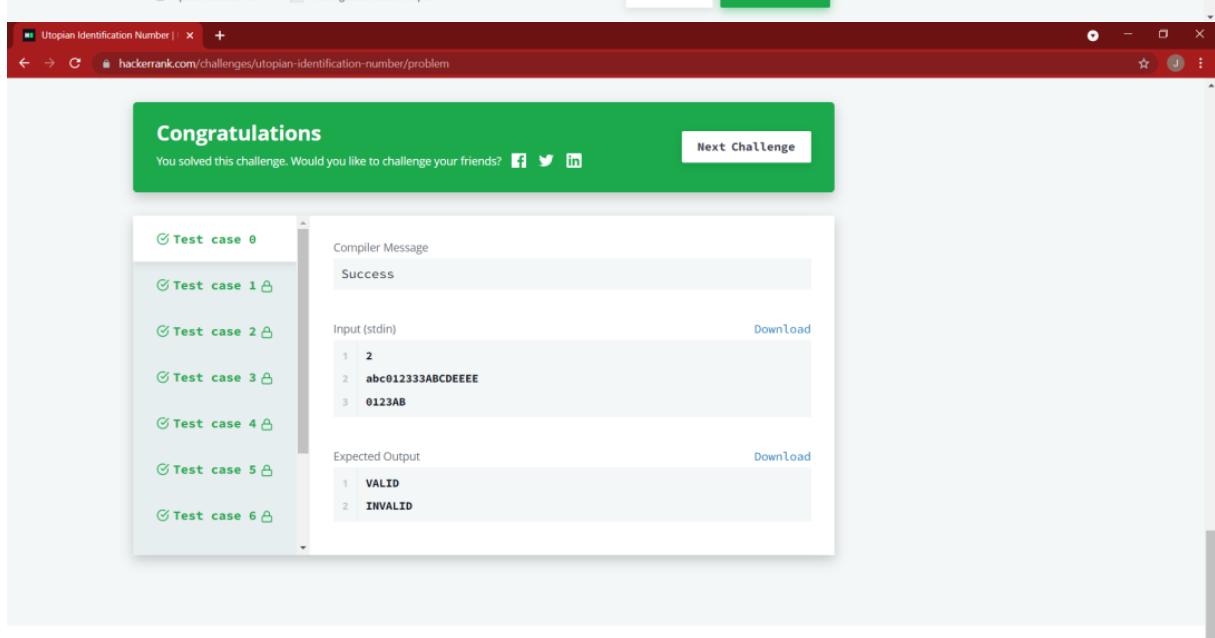
RATE THIS CHALLENGE

[MORE DETAILS](#)

The screenshot shows a code editor interface within a web browser window. The title bar reads "Utopian Identification Number | + | hackerank.com/challenges/utopian-identification-number/problem". The code area contains the following Python 3 script:

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 if __name__ == '__main__':
5     n = int(input())
6     Regex = r'^[a-z]{0,3}[0-9]{2,8}[A-Z]{3,}!'
7     for _ in range(n):
8         s = input()
9         if re.match(Regex, s):
10             print('VALID')
11         else:
12             print('INVALID')
```

Below the code editor, there are buttons for "Run Code" and "Submit Code". A status bar at the bottom indicates "Line: 12 Col: 29".



Saying Hi | HackerRank

hackerrank.com/challenges/saying-hi/problem

HackerRank NEW PREPARE CERTIFY COMPETE APPLY

Search jc8047

Prepare > Regex > Applications > Saying Hi

Saying Hi ★

Points: 200 Rank: 23402

Your Saying Hi submission got 15.00 points. Share Tweet

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Discussions

Given a sentence, *s*, write a RegEx to match the following criteria:

1. The first character must be the letter *H* or *I*.
2. The second character must be the letter *I* or *i*.
3. The third character must be a single space (i.e.: `\s`).
4. The fourth character must not be the letter *D* or *d*.

Given *n* lines of sentences as input, print each sentence matching your RegEx on a new line.

Input Format

The first line contains an integer, *n*, denoting the number of lines of sentences.
Each of the *n* subsequent lines contains some sentence *s* you must match.

Constraints

Author dheeraj
Difficulty Easy
Max Score 15
Submitted By 10257

NEED HELP? View discussions View top submissions RATE THIS CHALLENGE ★ ★ ★ ★ ★ MORE DETAILS

Change Theme Language Python 3

```
1 # Enter your code here. Read input from STDIN. Print output to STDOUT
2 import re
3
4 if __name__ == '__main__':
5     n = int(input())
6     r = r'^[H|h][I|i]\s[^D|d]'
7     for _ in range(n):
8         s = input()
9         if re.match(r, s):
10             print(s)
```

Line: 10 Col: 21

Saying Hi | HackerRank

hackerrank.com/challenges/saying-hi/problem

Congratulations

You solved this challenge. Would you like to challenge your friends? [Facebook](#) [Twitter](#) [LinkedIn](#)

[Next Challenge](#)

Test case 0

Compiler Message
Success

Test case 1

Input (stdin) [Download](#)

```
1 5
2 Hi Alex how are you doing
3 hI dave how are you doing
4 Good by Alex
5 hidden agenda
6 Alex greeted Martha by saying Hi Martha
```

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Expected Output [Download](#)

```
1 Hi Alex how are you doing
```

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature



JAYANT CHOUBEY