Ques1 Construct a turing machine over an alphabet {0, 1, #} where 0 indicates Blank which takes a non-null string of 1's and #'s and transfers the right most symbol to the left hand end. Thus ..... 000#1#1000 becomes 0001#1#1#000.
The Head is initially at the leftmost non blank symbol.

Turing machine to move rightmost symbol to left end
Alphabet : {0, 1, #} where 0 = blank
Initial Head Position: Leftmost non-blank symbol

Logic:
The TM scans right to find the rightmost symbol, erases it moves to the leftmost symbol, and shifts each symbol one position right while placing the erased symbol at the leftmost end.

Example : 000#1#1000 → 0001#1#1#000

7-tuple:
$M = (Q, \Sigma, \Gamma, \delta, q_0, 0, F)$ where
$\Sigma = \{1, \#\}$, $\Gamma = \{0, 1, \#\}$
$Q = \{q_0, q_R, q_{L1}, q_{L\#}, q_{S1}, q_{S\#}, q_{acc}\}$
Start state = q_0, Blank = 0, Final = $\{q_{acc}\}$

SKYSHOTS Spiral
Scanned with OKEN Scanner

Date.....

Transition Function (δ):

Current State	Read	Write	Move	Next State	Meaning
q_0	1 / #	same	R	q_R	Move right to find end
q_R	1 / #	same	R	q_R	Continue right
q_R	0	0	L	$q_{L?}$	Step back to last symbol
q_{L1}	1 / #	same	L	q_{L1}	Move to leftmost nonblank
q_{L1}	0	0	R	q_{S1}	Start shifting (carrying 1)
$q_{L\#}$	1 / #	same	L	$q_{L\#}$	Move left
$q_{L\#}$	0	0	R	$q_{S\#}$	Start shifting (carrying #)
q_{S1}	1	1	R	q_{S1}	Continue shift
q_{S1}	#	1	R	$q_{S\#}$	Swap carried symbol
q_{S1}	0	1	R	q_{acc}	Finish

SKYSHOTS Spiral

Date.....

q_1 S#	#	#	R	q_1 S#	Continue shift
q_1 S#	1	#	R	q_1 S 1	Swap
q_1 S#	0	#	R	qacc	Finish

Output : Rightmost symbol successfully moved to the left end..

Eg :
I/P → 000#1#1000
O/P → 0001# 1 # 1 # 000

Ques2 Design a Turing machine that accepts $\{0^n 1^n \mid n \geq 1\}$

TM (7 - tuple)
$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ where
- $\Sigma = \{0, 1\}$ (input alphabet)
- B = blank symbol
- $\Gamma = \{0, 1, X, Y, B\}$
- $Q = \{q_0, q_1, q_2, q_{check}, q_{acc}, q_{rej}\}$
- Start state q_0, Final $F = \{q_{acc}\}$.

Transition rules (δ) — table form
Interpretation: R = right , L = left.

State q_0 (find leftmost unmarked 0):

- $(q_0, 0) \rightarrow (q_1, X, R)$
- $(q_0, X) \rightarrow (q_0, X, R)$

SKYSHOTS
Spiral
Scanned with OKEN Scanner

- $(q_0, Y) \rightarrow (q_0, Y, R)$
- $(q_0, 1) \rightarrow (q_{-\{rej\}}, 1, R)$
- $(q_0, B) \rightarrow (q_{-\{check\}}, B, R)$

State q_1 (scan right to find an unmarked 1):

- $(q_1, X) \rightarrow (q_1, X, R)$
- $(q_1, Y) \rightarrow (q_1, Y, R)$
- $(q_1, 0) \rightarrow (q_1, 0, R)$
- $(q_1, 1) \rightarrow (q_2, Y, L)$
- $(q_1, B) \rightarrow (q_{-\{rej\}}, B, R)$

State q_2 (move left back to beginning to find next 0):

- $(q_2, X) \rightarrow (q_0, X, R)$
- $(q_2, 0) \rightarrow (q_2, 0, L)$
- $(q_2, 1) \rightarrow (q_2, 1, L)$
- $(q_2, Y) \rightarrow (q_2, Y, L)$
- $(q_2, B) \rightarrow (q_0, B, R)$

State q_{check} (verify all symbols are marked):

- $(q_{-\{check\}}, X) \rightarrow (q_{-\{check\}}, X, R)$
- $(q_{-\{check\}}, Y) \rightarrow (q_{-\{check\}}, Y, R)$
- $(q_{-\{check\}}, B) \rightarrow (q_{-\{acc\}}, B, R)$
- $(q_{-\{check\}}, 0) \rightarrow (q_{-\{rej\}}, 0, R)$
- $(q_{-\{check\}}, 1) \rightarrow (q_{-\{rej\}}, 1, R)$

q_{acc} and q_{rej} are halting accept / reject states

Spira

EXYSHOTS

Scanned with OKEN Scanner

Date.....

Ques3 Construct a Turing machine that could recognize the following Language $L = \{a^n b^n c^n \mid n \geq 1\}$

Given : Alphabet $\Sigma = \{a, b, c\}$
Tape Symbols : $\{a, b, c, X, Y, Z, B\}$

TM : $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ where
$Q = \{q_0, q_1, q_2, q_3, q\text{-check}, q\text{-acc}, q\text{-rej}\}$
Start state = q_0, Blank = B, Final = $\{q\text{-acc}\}$

Transition Table

Current state	Read	Write	Move	Next State
q_0	a	X	R	q_1
q_1	b	Y	R	q_2
q_2	c	Z	L	q_3
q_3	X/Y/Z	same	L	q_3
q_3	B	B	R	q_0
q_0	B	B	R	q-check
q-check	X/Y/Z	same	R	q-check
q-check	B	B	R	q-acc

Working: Each cycle replaces one a, b, c by X, Y, Z respectively.
If all symbols are correctly marked — Accept, else Reject.
Hence, TM accepts all strings of the form $a^n b^n c^n (n \geq 1)$

SKYSHOTS Spiral

Date.....

Quest Design a turing machine that compute 2's complement of given string over the $\Sigma = \{0, 1\}$. Show the output of your machine for string "0 00 00".

I/P : Binary string on tape (e.g., 0000)
O/P : 2's complement of the string

Logic :
1. Move head to the rightmost bit.
2. Copy bits as they are until the first 1 is found (leave that 1 as it is).
3. After that, complement all remaining bits ($0 \to 1, 1 \to 0$).

TM Definition :
$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ where
$\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, B\}$,
$Q = \{q_0, q_1, q_2, q_acc\}$, Start $= q_0$, Final $= q_acc$

Transition table :

Current State	Read	Write	Move	Next State
q_0	0/1	same	R	q_0
q_0	B	B	L	q_1
q_1	0	1	L	q_2
q_1	1	1	L	q_2
q_2	0	1	L	q_2
q_2	1	0	L	q_2
q_2	B	B	R	q_acc

SCRIBNOTS
Spiral
Scanned with OKEN Scanner

Date.....

Eg :
Input → 00000
Steps :
1. Rightmost 0 found, flip all bits → 11111
Output : 11111

Ques5 Which of the following is / are undecidable ?

a) G is a CFG. Is $L(G) = \phi$
b) G is a CFG. Is $L(G) = \Sigma^*$
c) M is a Turing Machine. Is L(M) is regular
d) A is a DFA and N is a NFA. Is L(A) = L(N)

(b) and (c) are Undecidable

Ques6 What is the solution for the following system of post correspondence problem ?
A = {100, 0, 1} B = {1, 100, 00}

We need to find a sequence of indices $(i_1, i_2, \ldots i_k)$ such that $A[i_1]\,A[i_2] \ldots A[i_k] = B[i_1]\,B[i_2] \ldots B[i_k]$

Step-by-step check :

Index	A [i]	B [i]
1	100	1
2	0	100
3	1	00

SKYSHOTS Spiral

Date.....

Try Sequence (1, 2, 3):
A → 100 0 1 = 10001
B → 1 100 00 = 110000 → not equal

Try sequence (1, 2, 1, 3):
A → 100 0 100 1 = 10001001
B → 1 100 1 00 = 1100100 → not equal

Try sequence (2, 1, 3):
A → 0 100 1 = 01001
B → 100 1 00 = 100100 → not equal

No combination makes A and B strings equal.

∴ Hence, the given PCP instance has no solution.

Ques Design TM, which reads an input and starts inverting 0's to 1's till the first 1. The first 1 also inverted. After it has inverted first 1 it read the next symbols and keeps them as they are till the next 1. After encountering 1, it starts repeating the cycle by inverting the symbol till next 1. It halts when it encounters a blank symbol?

Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ where
$Q = \{q_0, q_1, q_2, q_3, q_h\}$,
$\Sigma = \{0, 1\}$,
$\Gamma = \{0, 1, B\}$,
B = Blank symbol,
$F = \{q_h\}$.

Spiral
SKYSHOTS
Scanned with OKEN Scanner

Date.....

Transition function δ:

$(q_0, 0) \rightarrow (q_0, 1, R)$
$(q_0, 1) \rightarrow (q_1, 0, R)$
$(q_1, 0) \rightarrow (q_1, 0, R)$
$(q_1, 1) \rightarrow (q_0, 0, R)$
$(q_0, B) \rightarrow (q_h, B, N)$
$(q_1, B) \rightarrow (q_h, B, N)$

The TM alternates between two states — inverting 0's to 1's till it encounters a 1 and then keeping symbols unchanged till the next 1. It halts on blank.

Ques8 Prove that PCP problem is undecidable and explain with an example.

The post correspondence Problem (PCP) is undecidable.

Proof:
We reduce the Turing Machine acceptance problem (A_{tm}) to PCP. For a TM M and input w, we construct lists A and B such that a match (concatenation of same strings) exists iff M accepts w.
Hence, if PCP were decidable, we could decide A_{tm}, which is impossible.
$\therefore$ PCP is undecidable.

Eg:
Let
$A = \{1, 0\}$,
$B = \{0, 1\}$.

SKYSHOTS
Spiral
Scanned with OKEN Scanner

Date.....

No matching sequence → no solution
This shows how PCP checks matching patterns, but no general algorithm exists for all cases.

Conclusion:
PCP is undecidable, proved by reduction from the Turing Machine acceptance problem.

Ques9 Prove that the universal language L_u is recursively enumerable but not recursive. Also prove that the language L_d is not recursive or recursively enumerable.

1. Universal Language L_u:
$L_u = \{ \langle M, w \rangle \mid M \text{ accepts } w \}$

- A Turing Machine can simulate M on w, and if M accepts w, the TM halts and accepts.
→ Hence, L_u is Recursively Enumerable (RE).

- But the Halting Problem is reducible to L_u; if L_u were recursive, the Halting Problem would also be recursive, which is false.
→ Therefore, L_u is not recursive.

2. Diagonal Language L_d:
$L_d = \{ \langle M_i \rangle \mid M_i \text{ does not accept } w_i \}$
- L_d is the complement of L_u.
- Since L_u is RE but not recursive, its complement

Spiral

Scanned with OKEN Scanner

Date.....

L_d is neither recursive nor RE.

Conclusion:
L_u : Recursively Enumerable but not Recursive
L_d : Neither Recursive nor Recursively Enumerable.

Ques10 Construct a Turing Machine that accepts the language $a^n b^n c^n$.

Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ that accepts $L = \{a^n b^n c^n \mid n \geq 1\}$.

- $\Sigma = \{a, b, c\}$, blank B
- $\Gamma = \{a, b, c, X, Y, Z, B\}$ where X, Y, Z are marks
- $Q = \{q_0, q_1, q_2, q_3, q_{check}, q_{acc}, q_{rej}\}$
- Start state q_0, final $F = \{q_{acc}\}$

Transitions (δ) - write as (state, read) → (write, move next):

1. Find and mark leftmost unmarked a (in q_0):

$(q_0, a) \rightarrow (X, R, q_1)$
$(q_0, X) \rightarrow (X, R, q_0)$
$(q_0, Y) \rightarrow (Y, R, q_0)$
$(q_0, Z) \rightarrow (Z, R, q_0)$
$(q_0, B) \rightarrow (B, R, q\text{-check})$

2. In q_1 scan right to find leftmost unmarked b

SKYSHOTS
Spiral
Scanned with OKEN Scanner

Date.....

and mark it Y:

$(q_1, a) \rightarrow (a, R, q_1)$
$(q_1, X) \rightarrow (X, R, q_1)$
$(q_1, Y) \rightarrow (Y, R, q_1)$
$(q_1, b) \rightarrow (Y, R, q_2)$
$(q_1, B) \rightarrow (B, R, q_rej)$

3. In q_2 scan right to find leftmost unmarked c and mark it Z:

$(q_2, a/b/X/Y) \rightarrow (same, R, q_2)$
$(q_2, c) \rightarrow (Z, L, q_3)$
$(q_2, B) \rightarrow (B, R, q_rej)$

4. In q_3, move left to tape start (left of first nonblank) to begin next round:

$(q_3, a/b/c/X/Y/Z) \rightarrow (same, L, q_3)$
$(q_3, B) \rightarrow (B, R, q_0)$

5. In q_check verify all symbols are marked (only X, Y, Z allowed):

$(q_check, X/Y/Z) \rightarrow (same, R, q_check)$
$(q_check, B) \rightarrow (B, R, q_cc)$
$(q_check, a/b/c) \rightarrow (same, R, q_rej)$

Ques 11 Construct a Turing Machine to ~~prope~~ perform proper

Date.....

Subtraction.

Language: Input of the form $1^m \# 1^n$.
O/P: 1^{m-n} if $m > n$, otherwise blank.

TM (7-tuple):
$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_f\}$
- $\Sigma = \{1, \#\}$
- $\Gamma = \{1, X, Y, \#, B\}$
- Start q_0, Blank B, Final q_f.

Steps for transition:
1) In q_0: mark one 1 on the left ($1 \to X$), move right to find a matching 1 on the right.
2) In q_1: on first unmarked right 1, change it to Y, return left.
3) Repeat until no right 1 remains.
4) If right side ends first → restore all $X \to 1$, erase all Y and # → result 1^{m-n}.
5) If left side ends first → erase entire tape (result blank)

Key transitions:

- $(q_0, 1) \to (X, R, q_1)$
- $(q_1, 1) \to (Y, L, q_2)$
- $(q_2, X) \to (X, L, q_2)$
- $(q_2, B) \to (B, R, q_0)$
- $(q_1, B) \to (B, L, q_3)$ (right 1's finished)

Spiral

Scanned with OKEN Scanner

Date.....

- $(q_3, X) \rightarrow (1, R, q_3)$
- $(q_3, Y) \rightarrow (B, R, q_3)$
- $(q_3, \#) \rightarrow (B, R, q_3)$
- $(q_3, B) \rightarrow (B, R, q_f)$

O/P examples :

- I/P : 1111#11 → O/P : 11 (4-2)
- I/P : 11# 111 → O/P : blank (2-3)

Ques 12 Prove that the halting problem of Turing Machine over $\{0, 1\}^*$ is unsolvable.

The halting problem is :

$$HALT = \{\langle M, w\rangle \mid M \text{ halts on } w\}.$$

Assume it is decidable by a TM H.
Define a new TM D such that on input $\langle X\rangle$:

- If H says that X halts on $\langle X\rangle$, then D goes into an infinite loop.
- If H says that X does not halt on $\langle X\rangle$, then D halts.

Now run D on its own input $\langle D\rangle$.
This gives a contradiction :

- If D halts, then by definition it should loop.
- If D loops, then by definition it should halt.

Hence no such decider H can exist.
Therefore, the halting problem is unsolvable (undecidable).

SKYSHOTS

Spiral

Scanned with OKEN Scanner

Date.....

Ques13 Let $\Sigma = \{a, b\}^*$. Let A and B are the lists of three strings as given below:
$A = \{b, bab3, ba\}$ $B = \{b3, ba, a\}$
Does this instance of PCP have a solution? Justify your answer.

NO, this PCP instance has no solution.

For a solution, the concentration of A - strings and B - strings must become equal. Let index 1 = (b, bbb), index 2 = (babbb, ba), index 3 = (ba, a).

1. First - symbol mismatch:
All A_i start with b, but $B_3 = a$.
So index 3 cannot be first → otherwise the two strings start differently.

2. Last - symbol mismatch:
A_2 ends with b, B_2 ends with a.
So index 2 cannot be last.

3. Length feasibility
For any ~~length~~ solution, total lengths must satisfy:
$1 \cdot n_1 + 5 \cdot n_2 + 2 \cdot n_3 = 3 \cdot n_1 + 2 \cdot n_2 + 1 \cdot n_3$
$\Rightarrow 3n_2 + n_3 = 2n_1$

Testing all small combinations that satisfy this shows no sequence can match both the start / end constraints and produce identical concatenations.

SKYSHOTS
Spiral
Scanned with OKEN Scanner

Date.....

⇒ Since no valid sequence of indices can make the A-string equal to the B-string, this PCP instance has no solution.

Ques14 Prove L_{ne} is recursively enumerable.

Let Σ be the tape/input alphabet.
Define $L_{ne} = \{\langle M \rangle \mid L(M) \neq \phi\}$

To show L_{ne} is recursively enumerable, give a semi-decider S that on input $\langle M \rangle$ accepts iff M accepts some string.

Semi-decider S (dovetailing procedure):

1. Enumerate all strings $w_1, w_2, w_3, \ldots$ over Σ (e.g. in lexicographic order).
2. Run a dovetailed simulation of M on all w_i: for stage $t = 1, 2, 3, \ldots$ simulate one more step of M on each w_i for all $i \leq t$.
3. If during simulation some run $M(w_i)$ halts and accepts, then halt and accept.

Correctness:

- If $\langle M \rangle \in L_{ne}$, there exists some w with M accepting w. The dovetailed simulation will eventually simulate the accepting run and S accepts.
- If $\langle M \rangle \notin L_{ne}$, no simulation ever accepts, so S never accepts.

Spiral

SKYSHOTS

Scanned with OKEN Scanner

Date.....

Thus L_{ne} has a Turing machine that accepts exactly the members of L_{ne}, so L_{ne} is recursively enumerable.

Ques15 Prove that for two recursive languages L1 and L2 their union and intersection is recursive.

Let L_1 and L_2 be recursive.
So there exist deciders M_1 and M_2 that always halt and correctly decide L_1 and L_2.

1. Union $L_1 \cup L_2$
Construct TM M:

- On input x:
Run $M_1(x)$. If it accepts → accept
Else run $M_2(x)$. If it accepts → accept
Otherwise → reject.

Since both machines halt, M also halts.
Hence $L_1 \cup L_2$ is recursive.

2. Intersection $L_1 \cap L_2$
Construct TM N:

- On input x:
Run $M_1(x)$. If it rejects - reject.
Else run $M_2(x)$. If it accepts → accept, else reject.

Again both always halt → N halts. Hence $L_1 \cap L_2$ is recursive.

SKYSHOTS
Spiral
Scanned with OKEN Scanner

Date.....

Ques16 Define Post correspondence problem. Let $\Sigma = \{0, 1\}$. Let A and B be the lists of three strings, each defined as

	List A	List B
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

Does this PCP have a solution?

- Tile 3 has A = 10 and B = 0, so it cannot start (top begins with 1, bottom with 0 → mismatch).
- Tile 2 ends with 1 on top but 0 on bottom, so it cannot end any matching sequence.
- Trying possible combination starting with tile 1:

→ 1 → top = 1, bottom = 111 (prefix mismatch)

→ 1, 1 or 1, 3 or 1, 2, 3 etc, also fail because prefixes diverge.

No sequence makes A-side = B-side

$\boxed{\text{This PCP has no solution.}}$

Ques17 Does a PCP solution exist for the following set?

(10, 101), (01, 100), (0, 10), (100, 0), (1, 010)

- The first tile must have the same starting symbol on top and bottom. Only tile 1 (10, 101) qualifies → sol^n must start with tile 1.

Date.....

- But tile 1 already creates mismatch:

Top = 10

Bottom = 101

Bottom has an extra 1 which no other tile can remove or balance without causing further prefix mismatch.

- Trying possible short sequences starting with tile 1 (like 1-3, 1-4, 1-1-3, etc) all fail because the prefixes never match.

$\boxed{\text{This PCP instance has no solution.}}$

Ques19 Design a Turing machine for Reversing the given string {abb}.

TM marks each symbol from left (a→X, b→Y), moves to a separator #, writes the same symbol at the end, returns, and repeats. After all symbols are marked, it erases the original part & #.

Key Transitions:

1. Create #:

(q0, a/b → R), (q0, B → write #, L → q1)

2. Mark next symbol:

(q1, a → X, R, q2), (q1, b → Y, R, q2)

3. Append after #:

Move to #, then write a or b to its right.

4. Cleanup:

Erase X, Y, # and halt.

O/P for input abb

Appending order = a, b, b → final reversed string:

$\boxed{\text{bba}}$

Date.....

Ques 21 Show that any PSPACE-hard language is also NPhard.

Proof:

1. NP ⊆ PSPACE
Every NP problem can be solved using polynomial space (because a nondeterministic polynomial-time TM uses only polynomial space).
2. Definition of PSPACE-hard
A language L is ~~space~~ PSPACE-hard if every language in PSPACE reduces to L using a polynomial-time reduction.
3. Since NP ⊆ PSPACE, every NP language is also in PSPACE.
4. Therefore, if all PSPACE languages reduce to L, then in particular all NP languages also reduce to L.

∴ Every NP problem reduces to a PSPACE-hard problem, every PSPACE-hard language is automatically NP-hard.
Hence proved.

Ques 25 Check the following is PCP or not.

I	A	B
1	00	0
2	001	11
3	1000	011

Try to build matching strings:

- Using tile 1: A = 00, B = 0 → mismatch.
- Extend with tile 1 again: A = 0000, B = 00 → still

SKYSHOTS
Spiral
Scanned with OKEN Scanner

Date.....

mismatch.

- Trying combinations with tiles 2 or 3 always leaves extra unmatched 0's / 1's on one side.
- No sequence can make the A-side and B-side equal.

$\boxed{\text{No PCP sol}^n \text{ exists for this set.}}$

Scanned with OKEN Scanner