Task 1 - Nash Equilibria and Subgame Perfect Equilibria

(a) The Nash Equilibria in a 2×2 Normal-Form Game
The given normal-form game is:

Mario \ Luigi	Boost	Jump
Throws	(-2, -8)	(1, 7)

Step taken as follows:
Step 1: Identify Best Responses

A Nash equilibrium is a strategy profile where no player can improve their payoff by unilaterally changing their strategy.

- Mario's Best Response to Luigi's Strategies:
  - If Luigi chooses Boost, Mario's options:
    - Throw → Payoff = -2
    - Jump → Payoff = 1 (Better choice)
    - Best Response: Jump
  - If Luigi chooses Jump, Mario's options:
    - Throw → Payoff = 1
    - Jump → Payoff = 7 (Better choice)
    - Best Response: Jump
- Luigi's Best Response to Mario's Strategies:
  - If Mario chooses Throw, Luigi's options:
    - Boost → Payoff = -8
    - Jump → Payoff = 7 (Better choice)
    - Best Response: Jump
  - If Mario chooses Jump, Luigi's options:
    - Boost → Payoff = 1
    - Jump → Payoff = 7 (Better choice)

    - Best Response: Jump

Step 2: Find Nash Equilibria

The best responses are:

- Mario plays Jump.
- Luigi plays Jump.

Thus, the unique Nash Equilibrium is (Jump, Jump) → (7,7).

(b) Subgame Perfect Equilibria (SPE) in an Extensive-Form Game (15%)
The game tree has:

- Player 1 (Root decision node) chooses between L, M, R.
- Player 2 makes decisions at each branch with actions A, B, C, D, E, F.
- Payoffs are provided for each terminal node.

Step 1: Use Backward Induction

We analyze the game starting from the terminal nodes and move backward.

1. At Node 2 (for L choice by Player 1):
    - Player 2 chooses between A (2,1) and B (1,2).
    - Best response: A → (2,1).
2. At Node 2 (for M choice by Player 1):
    - Player 2 chooses between C (2,3) and D (5,3).
    - Best response: D → (5,3).
3. At Node 2 (for R choice by Player 1):
    - Player 2 chooses between E (1,0) and F (6,-1).
    - Best response: E → (1,0).

Step 2: Player 1's Decision

Now, Player 1 compares the payoffs:

- Choosing L → (2,1)
- Choosing M → (5,3)
- Choosing R → (1,0)

Best response for Player 1: M (5,3) (since 5 > 2 and 5 > 1 for Player 1’s payoff).

Step 3: Identify the Subgame Perfect Equilibrium

- Player 1 chooses M.
- Player 2, at M’s node, chooses D.
- The SPE strategy profile is: (M, D) → (5,3).

Task 2 - Auction Simulation and Equilibrium Analysis

2a. Simulating the Auction Process in MATLAB (Please note that the multiple replicate is available in the .m file)

Understanding the Auction Mechanism

- Each of the 3 buyers submits a bid randomly chosen from [1, valuation].
- The two highest bidders win the item.
- The price paid is the third-highest bid.
- The profit for a winning buyer is valuation - payment.
- Social welfare is the sum of the winners' valuations.
- The process is repeated for 7 rounds and replicated 11 times.

```
% Number of rounds

rounds = 7;

% Number of buyers

num_buyers = 3;

% Define valuations of the buyers

valuations = [12, 9, 4];
```

```
% Initialize storage for results
bids_history = zeros(rounds, num_buyers);
profits = zeros(rounds, num_buyers);
social_welfare = zeros(rounds, 1);
% Simulate the auction for 7 rounds
for r = 1:rounds
    % Generate random bids for each buyer
    bids = [randi([1, 12]), randi([1, 9]),
  randi([1, 4])];
    bids_history(r, :) = bids;

    % Sort bids in descending order to determine
  winners
    [sorted_bids, indices] = sort(bids,
  'descend');

    % Two highest bidders win the item
    winner1 = indices(1);
    winner2 = indices(2);

    % The third-highest bid is the price
    price = sorted_bids(3);
```

```
    % Compute profits for each buyer
    for i = 1:num_buyers
        if i == winner1 || i == winner2
            profits(r, i) = valuations(i) -
   price;
        else
            profits(r, i) = 0;
        end
    end

    % Calculate social welfare (sum of valuations
   of winners)
    social_welfare(r) = valuations(winner1) +
   valuations(winner2);

    % Display results for the round
    fprintf('Round %d: Bids = [%d, %d, %d],
   Winners = [%d, %d], Price = %d\n', ...
        r, bids(1), bids(2), bids(3), winner1,
   winner2, price);
    fprintf('Profits: [%d, %d, %d]\n\n',
   profits(r, 1), profits(r, 2), profits(r, 3));
end
```

```
% Calculate total profits and overall social
  welfare

total_profits = sum(profits, 1);

overall_social_welfare = sum(social_welfare);

% Display final results

fprintf('Total Profits: [%d, %d, %d]\n',
  total_profits(1), total_profits(2),
  total_profits(3));

fprintf('Overall Social Welfare: %d\n',
  overall_social_welfare);
```

Replication 1	Vector of Bids	Winners(two buyers that receive an item.)	Profit of Each Buyer	Social Welfare
Round 1	[10, 9, 1]	[1, 2]	[11, 8, 0]	21
Round 2	[11, 6, 1]	[1, 2]	[11, 8, 0]	21
Round 3	[4, 5, 4]	[2, 1]	[8, 5, 0]	21
Round 4	[12, 2, 4]	[1, 3]	[10, 0, 2]	16
Round 5	[12, 5, 4]	[1,2]	[8, 5, 0]	21
Round 6	[2, 4, 4]	[2,3]	[0, 7, 2]	13
Round 7	[10, 9, 3]	[1,2]	[9, 6, 0]	21

Total(Rounds 1-7)			**[57, 39, 4]**	**134**

Replications	total_profits	overall_social_welfare
Replications 1	57, 39, 4;	134
Replications 2	59, 41, 4;	134
Replications 3	49, 40, 7;	126
Replications 4	51, 30, 13;	116
Replications 5	61, 50, 2;	139
Replications 6	67, 40, 1;	142
Replications 7	72, 30, 6;	132
Replications 8	63, 44, 5;	134
Replications 9	60, 50, 3;	139
Replications 10	54, 44, 3;	139
Replications 11	71, 37, 3	137

Average total profit for 11 replications	[60.36, 40.45, 4.27]
Average social welfare for 11 replications	133.64

2b. Using Brute Force Search method(Please note the .m file contain the results of the table format given in the question)

Understanding Equilibrium in the Auction

- A strategy (bid choice) is an equilibrium if no player can improve their total profit by changing their bid unilaterally.
- This requires checking all possible bid combinations and seeing if any buyer can benefit by deviating.

```
% Define buyer valuations

valuations = [12, 9, 4]; % v1 = 12, v2 = 9, v3 = 4


% Define bid range (each buyer bids between 1 and their valuation)

b1_range = 1:12;

b2_range = 1:9;

b3_range = 1:4;


% Initialize storage for equilibria

equilibria = [];

social_welfare_list = [];


% Brute force search through all possible bid combinations

for b1 = b1_range

    for b2 = b2_range
```

```
for b3 = b3_range

    % Store the current bid combination

    bids = [b1, b2, b3];



    % Sort bids in descending order to determine winners

    [sorted_bids, indices] = sort(bids, 'descend');



    % Two highest bidders win the item

    winner1 = indices(1);

    winner2 = indices(2);



    % The third-highest bid is the price

    price = sorted_bids(3);



    % Compute profits for each buyer

    profits = zeros(1, 3);

    for i = 1:3

        if i == winner1 || i == winner2

            profits(i) = valuations(i) - price;

        else

            profits(i) = 0;

        end
```

```
    end


    % Compute social welfare (sum of valuations of winners)

    social_welfare = valuations(winner1) + valuations(winner2);


    % Check if the bid is an equilibrium (no buyer can profit by
deviating)

    is_equilibrium = true;

    for i = 1:3

        for new_bid = 1:valuations(i)

            if new_bid ~= bids(i)

                % Simulate the effect of changing the bid

                new_bids = bids;

                new_bids(i) = new_bid;

                [new_sorted_bids, new_indices] = sort(new_bids,
'descend');

                new_winner1 = new_indices(1);

                new_winner2 = new_indices(2);

                new_price = new_sorted_bids(3);


                % Compute new profit

                new_profits = zeros(1, 3);

                for j = 1:3
```

```
                if j == new_winner1 || j == new_winner2
                    new_profits(j) = valuations(j) - new_price;
                else
                    new_profits(j) = 0;
                end
            end

            % If the buyer improves their profit by deviating, it's not
an equilibrium
            if new_profits(i) > profits(i)
                is_equilibrium = false;
                break;
            end
        end
    end
    if ~is_equilibrium
        break;
    end
end

% If equilibrium is found, store it
if is_equilibrium
    equilibria = [equilibria; b1, b2, b3, social_welfare];
```

```
                social_welfare_list = [social_welfare_list, social_welfare];
            end
        end
    end
end

% Compute average social welfare for all equilibria
average_social_welfare_equilibria = mean(social_welfare_list);

% Display equilibria
fprintf('Buyer 1\tBuyer 2\tBuyer 3\tSocial Welfare\n');
for i = 1:size(equilibria, 1)
    fprintf('%d\t%d\t%d\t%d\n', equilibria(i, 1), equilibria(i, 2), equilibria(i,
     3), equilibria(i, 4));
end

% Display average social welfare
fprintf('Average Social Welfare for All Equilibria: %.2f\n',
     average_social_welfare_equilibria);
```

Average social welfare for all equilibria	21.00

Task 3 - Cournot and Stackelberg Duopoly Analysis

3a Cournot Duopoly Model

Step 1: We need to define the Profit Functions

The inverse demand function (price as a function of total quantity) is given by:

$$P(q1,q2) = 43 - 3\,(q1+q2)$$

The **cost functions** for Firm 1 and Firm 2 are:

$$c1(q1)= 5 + 4q1\ ,\ c2(q2)$$
$$=10 + q2$$
$$101_2$$

Each firm's **profit function** is given by:

$$\pi 1(q1,q2)=P(q1,q2)\cdot q1-c1(q1)$$
$$\pi 2(q1,q2)=P(q1,q2)\cdot q2-c2(q2)$$

Expanding:

$$\pi 1(q1,q2)=(43-3(q1+q2))q1-(5+4q1)$$
$$=43q1-3q_1^2-3q_1q_2-5-4q_1$$
$$=39q_1-3q_1^2-3q_1q_2-5$$

$$\pi 2(q_1,q_2)=(43-3(q_1+q_2))q_2-(10+q_2)$$
$$=43q_2-3q_2^2-3q_1q_2-10-q_2$$
$$=42q_2-3q_2^2-3q_1q_2-10$$

Step 2: We need to find the Nash Equilibrium

Each firm maximizes its profit by solving:

Firm 1's Best Response

$$\partial \pi_1/\partial q_1 = 39 - 6q_1 - 3q_2 = 0$$

Solving for q1: $q1 = 39 - 3q_2 \; / \; 6$

$= 6.5 - 0.5q2$

Firm 2's Best Response

$$\partial \pi_2/\partial q_2 = 42 - 6q_2 - 3q_1 = 0$$

Solving for q_2:

$$q_2 = 42 - 3q_1 / 6$$

$$= 7 - 0.5q_1$$

Solve the System of Equations

$$q_1 = 6.5 - 0.5q_2$$

$$q_2 = 7 - 0.5q_1$$

Substituting q2 into q1:

$$q_1 = 6.5 - 0.5(7 - 0.5q1)$$

$$q_1 = 6.5 - 3.5 + 0.25q1$$

$$0.75q_1 = 3$$

$$q_1 = 4$$

Substituting $q_1 = 4$ into $q_2 = 7 - 0.5q_1$:

$$q_12 = 7 - 0.5(4) = 5$$

Step 3: Compute Profits, Consumer Surplus, and Total Surplus

Using $q_1=4$, $q_2=5$:

$$P = 43 - 3(4+5) = 43 - 27 = 16$$

$$\pi_1 = (16\times4) - (5+4\times4) = 64 - 21 = 43$$

$$\pi_2 = (16\times5) - (10+5) = 80 - 15 = 65$$

Consumer Surplus: $CS = {}^1/_2(43-16)/2\times9$

$= 227\times9$

$= 121.5$

Total Surplus: $TS = \pi_1 + \pi_2 + CS = 43 + 65 + 121.5 = 229.5$

Cournot Equilibrium	Value
q_1	4
q_2	5
$\pi_1(q_1,q_2)$	43
$\pi_2\,(q_1,q_2)$	65
Consumer Surplus (CS)	121.5
Total Surplus (TS)	229.5

(b) Stackelberg Leader-Follower Model (15%)

In this model:

- Firm 1 is the **leader**, choosing q_1 first.
- Firm 2 is the **follower**, reacting using the **reaction function**:

$$r_2(q_1) = 5 + 3q_1$$

Step 1: Leader's Profit Function

$$\pi 1=(43-3(q1+q2))q1-(5+4q1)$$

Substituting q2=5+3q1:

$$\pi_1=(43-3(q1+5+3q_1))q_1-(5+4q_1)$$

$$\pi_1=(43-3(4q_1+5))q_1-5-4q_1$$

$$\pi_1=(43-12q_1-15)q_1-5-4q_1$$

$$\pi_1=(28-12q_1)q_1-5-4q_1$$

$$\pi_1=28q_1-12q_1^2-5-4q_1$$

$$\pi_1= 24q_1-12q_1^2-5$$

Step 2: Solve for Stackelberg Equilibrium

Maximizing π_1: $d\pi_1/dq_1 = 24-24q_1 = 0$

$$q_1 = 1$$

Using $r_2(q_1)$: $q_2 = 5 + 3\,(1) = 8$

Leader's Profit: $\pi_1 = (16\times1)-(5+4)=16-9=7$

Stackelberg Equilibrium	Value
q_1	1
q_2	8
$\pi_1(q_1, q_2)$	7

(c) Collusion Models

1. Joint Profit Maximization: Firms act as a monopoly, splitting profit to maximize joint revenue. Profit is maximized at

$q_1+q_2 = 7.$

Each firm sets lower output for a higher price.

2. Price-Fixing Agreement: Firms agree on a minimum price, reducing competition and increasing profits.

Is collusion profitable?
Yes, if **enforceable**, but it risks detection and legal consequences.