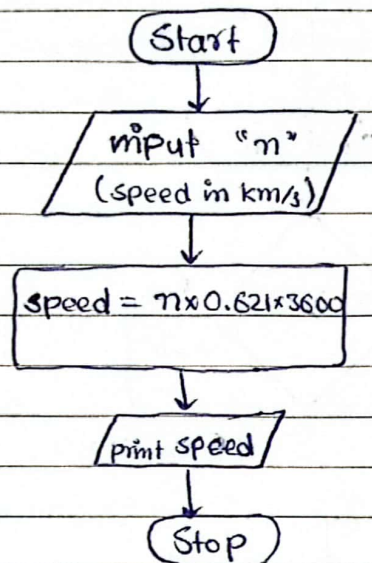1) Design the flowchart to convert kilometers per seconds to miles per hour.

```
                    ( Start )
                        |
                        v
               / Input  "n"        /
              / (speed in km/s)   /
                        |
                        v
          [ speed = n×0.621×3600 ]
                        |
                        v
                /print speed/
                        |
                        v
                  ( Stop )
```
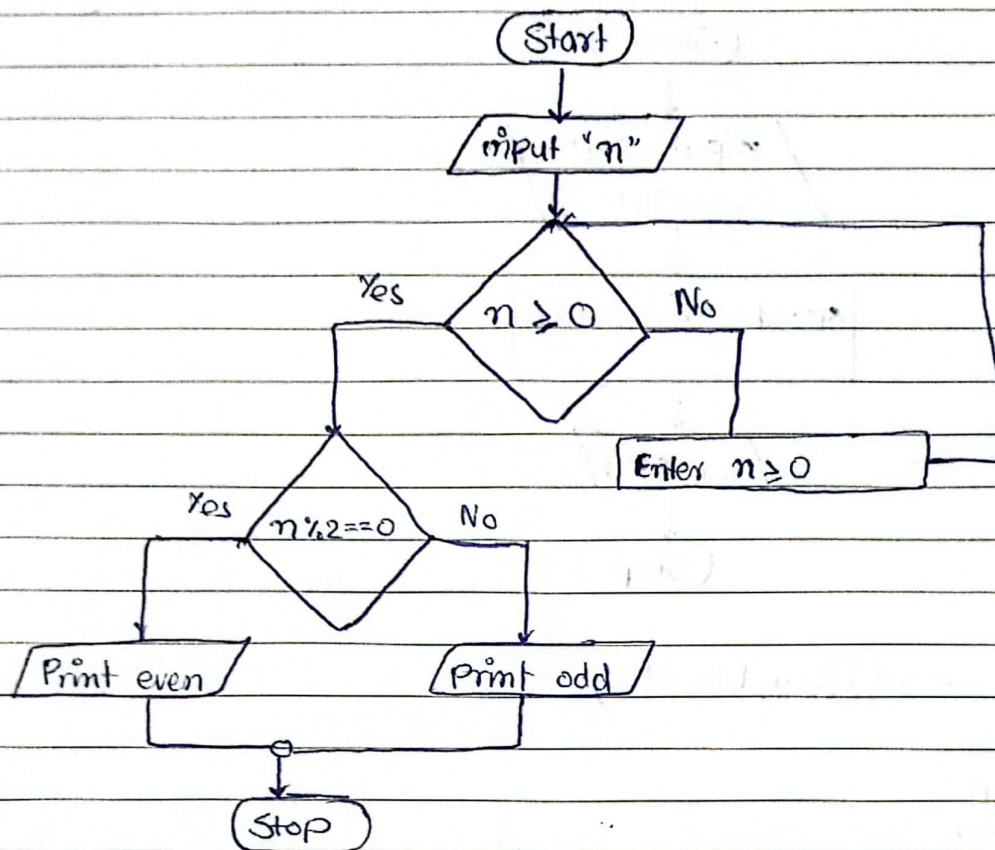
~(Pseudo code)~

1. Start
2. Input "n" (Speed in km/s)
3. Calculate speed
   speed = n × 0.621 × 3600
4. print speed
5. Stop.

~(IPO)~

Input	Processing	Module Reprence	Output
"n" (Speed in km/s)	Enter input 'n' Calc: speed speed = n×0.621×3600 print speed End	Read Calculate Print PayRoll Controll	speed in miles /h

2) Design a flowchart that will ask for a positive number greater than 0 and prints whether a number is even or odd

Start

input "n"

$n \geq 0$

Yes   No

Enter $n \geq 0$

$n \% 2 == 0$

Yes   No

Print even

Print odd

Stop

⟨ Pseudocode ⟩

1. Start
2. Input n
3. If   n >= 0
    If   n%2 == 0
     Print even
   Else
     Print odd
4. Else
    print " enter n >= 0
5. Stop

Input	Processing	Module Refrence	Output
I "n" (number)	Enter n Check n%= 0 Check n%2==0 Print result End	Read Check condition Check condition Print PayRoll Controll	even or odd

Q3: Design a flowchart for both provinces (Sindh & Punjab) to make this decision easy by asking relevant information from the user and giving output whether it is legal to marry to or not.
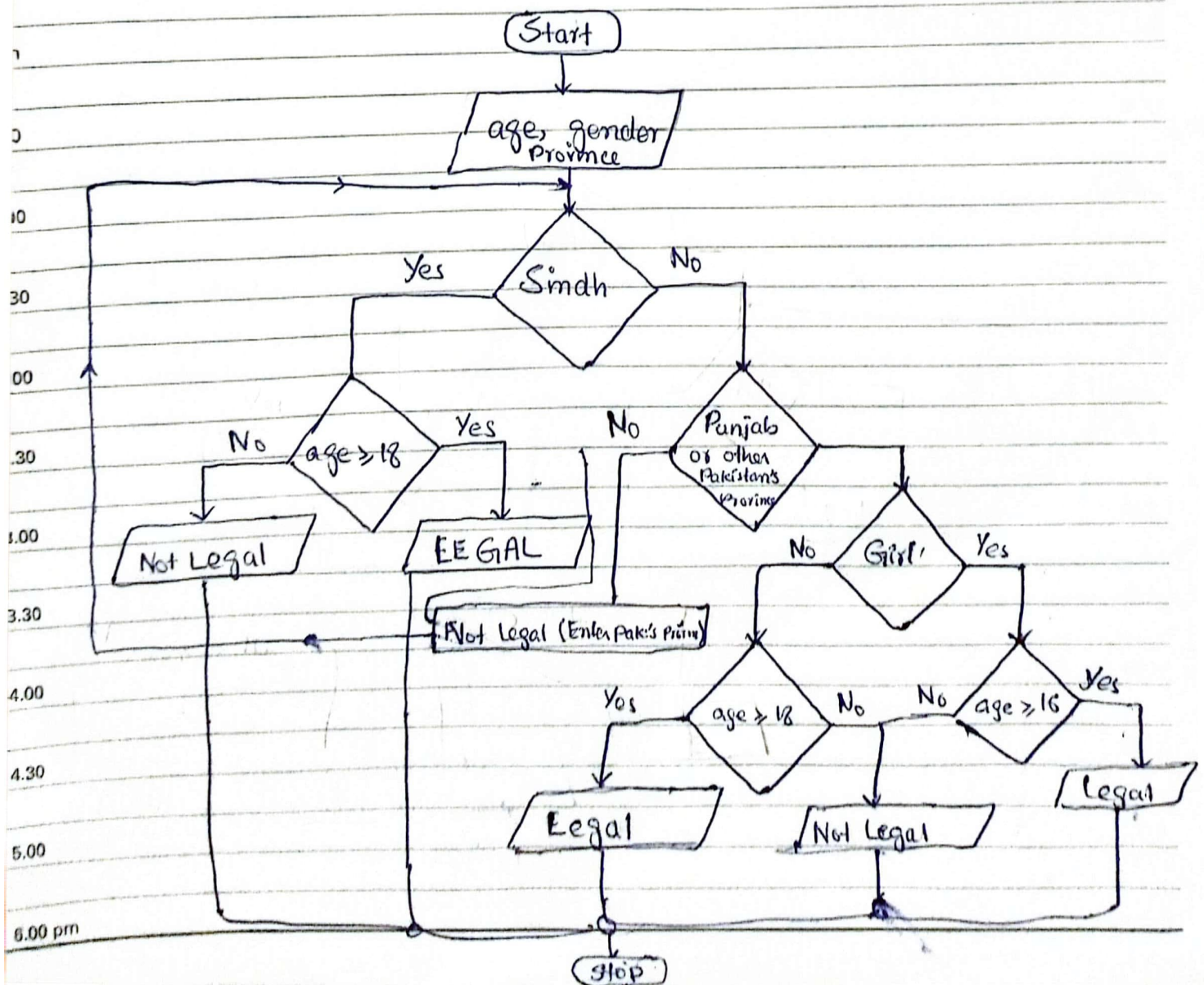
〜❦ Pseudocode ❦〜

1. Start
2. Input age, gender, province
3. If Province is sindh
       If age ≥ 18
       Print legal
   Else
       Print Not legal
4. Else if
       Province is Punjab & other
   If Gist
       If age ≥ 16
       Print legal
       Else not legal
5. Else
       If age ≥ 18
       Print Legal
   Else
       Print not legal
6. Else if
       Print Not data found (Enter pakisten's psovince)
7. Stop

〜❦ IPO ❦〜

Input	Processing	Module Refrence	Output
	Enter age	Read	
age	Enter Grender	Road	Legal &
Grender	Enter province	Read	Not Legal to
Province	Check province	Calculate	Marry
	Check age	Print	
	Check gender	Pay Roll Controll	
	Print output		
	Stop		

4) Design a flowchart to help Mr Bhoola in Calculating mony for shopping

```
                        ┌─────────┐
                        │  Start  │
                        └────┬────┘
                             │
                        ╱────┴────╲
                        │  Money   │
                        ╲────┬────╱
                             │
              ┌──────────────┴──────────────┐
              │  onion 1kg = P1              │
              │  apricots 1kg = P2           │
              │  grapes  1kg = P3            │
              │  tometoes 1kg = P4           │
              └──────────────┬──────────────┘
                             │
                        ╱────┴────╲
                        │ Quantity │
                        ╲────┬────╱
                             │
         ┌───────────────────┴───────────────────┐
         │  Onion = Quantity × P1                 │
         │  Apricots = Quantity × P2              │
         │  Grapes = Quantity × P3                │
         │  tomatoes = Quantity × P4              │
         │  Total Price = onion + Apricots + grapes + tomatoes │
         └───────────────────┬───────────────────┘
```

$$\text{Onion} = \text{Quantity} \times P_1$$
$$\text{Apricots} = \text{Quantity} \times P_2$$
$$\text{Grapes} = \text{Quantity} \times P_3$$
$$\text{tomatoes} = \text{Quantity} \times P_4$$
$$\text{Total Price} = \text{onion} + \text{Apricots} + \text{grapes} + \text{tomatoes}$$

Decision: money ≥ Total Price

- Yes →
- No → Money is not Sufficient → Money need = Total Price - Money → Pay full and

Decision: money = Total Price

- Yes → Thanks sufficient money
- No → Change = Money - Total Price → Print Change

$$\text{Money need} = \text{Total Price} - \text{Money}$$
$$\text{Change} = \text{Money} - \text{Total Price}$$

Stop

⊰ Pseudocode ⊱

1) Start
2. Input Money
3. Declare
    onion   1kg = P₁
    apricots 1kg = P₂
    grapes 1kg = P₃
    tometoes 1kg = P₄
4. Input Quantity
5. Calculating price
    Onion = Quantity × P₁
    Apricots = Quantity × P₂
    Grapes = Quantity × P₃
    tometoes = Quantity × P₄
6) Calculating total price
    Total price = Onion + Apricots + Grapes + tometoes
7. If
   money ≥ Total price
        If
            money == total price
            Thanks you change is 0
        Else
            Change = Money - Total price
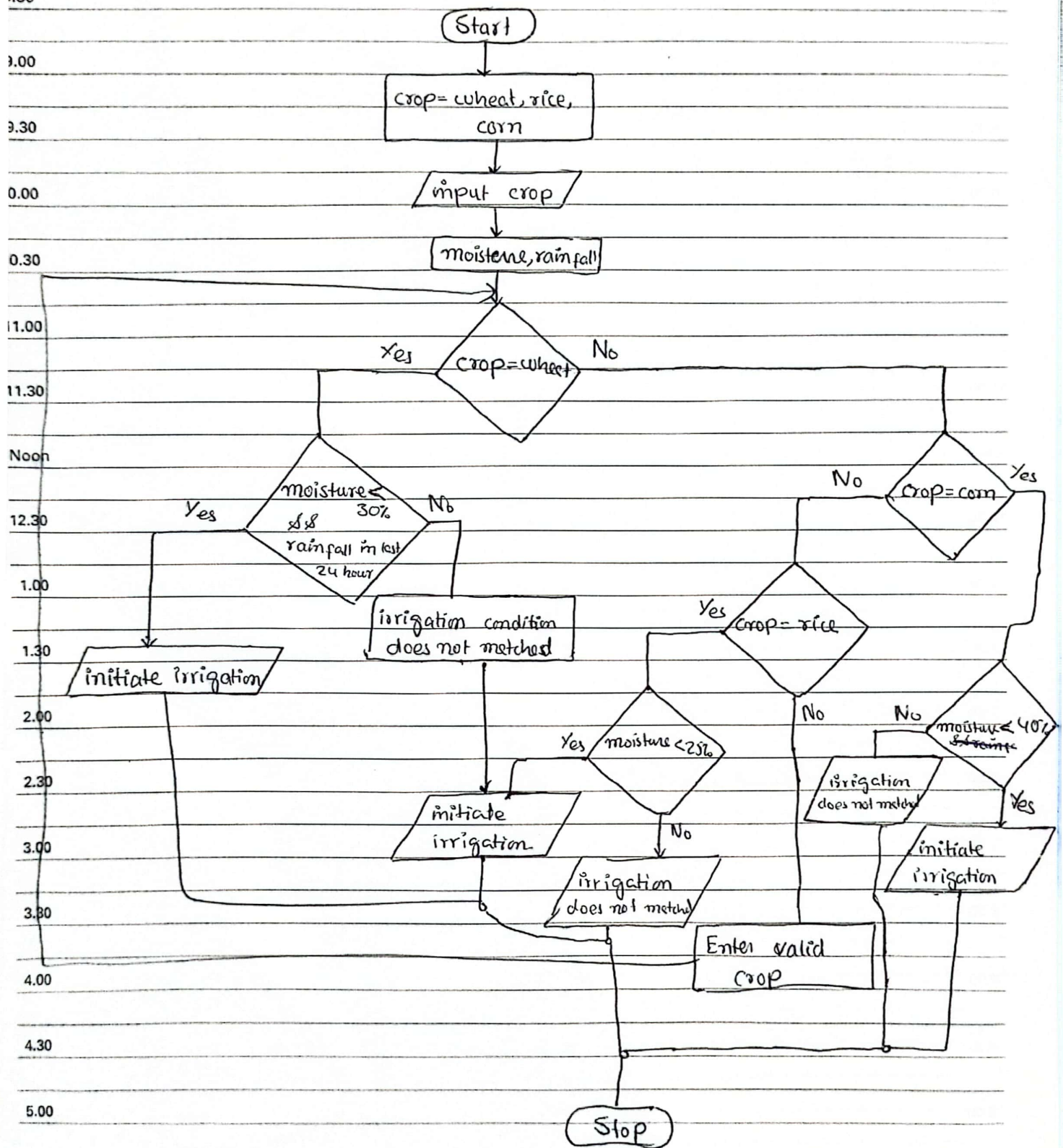   Print change
    Else
        Pay full money
8. End

⊰ IPO ⊱

Input	Processing	Module Refrence	Output
	Enter money	Read	
Money	Declaring variable		
	Enter Quantity	Read	Change or
	Calculating price	Calculate	eligibility of money to
	Calculating total price	Calculate	Shopping
	Condition of greater	Print 0 change	
	Condition of equal price	Print x change	
	End	Payroll Control	

5) Crop Management

```
                    ┌──────────┐
                    │  Start   │
                    └────┬─────┘
                         │
              ┌──────────▼──────────┐
              │ crop= wheat, rice,  │
              │       corn          │
              └──────────┬──────────┘
                         │
                  ┌──────▼──────┐
                 / input crop  /
                 └──────┬──────┘
                        │
              ┌─────────▼─────────┐
              │ moisture, rainfall │
              └─────────┬─────────┘
                        │
```

Decision: crop = wheat?
- Yes → moisture < 30% && rainfall in last 24 hour
  - Yes → initiate irrigation
  - No → irrigation condition does not matched
- No → crop = corn?
  - Yes → crop = rice?
    - Yes → moisture < 25%
      - Yes → initiate irrigation
      - No → irrigation does not matched
    - No → moisture < 40% && rainfall
      - Yes → initiate irrigation
      - No → irrigation does not matched
  - No → Enter valid crop

```
                  ┌──────────┐
                  │   Stop   │
                  └──────────┘
```

—❧ Pseudocode ❧—

1. Start
2. Initiate crop = wheat, rice, corn
3. Input crop
4. Initiate moisture, rainfal
5. If
    crop = wheat
      If moisture < 30% && rainfall in last 24hour
    initiate irrigation
6. Else if
    crop = corn
    moisture < 40%
    initiate irrigation
7. Else if
    crop = rice
    moisture < 25%
    initiate irrigation
8. Else
    print "Irrigation condition does not matched
9. End

—❧ IPO ❧—

Input	Processing	Module Reference	Output
	Enter crop	Read	
crop	(crop = wheat, corn, rice)	Check if condition meets	
	Initiate moisture, rainfall	Print	Initiation of
	Condition moisture & rainfall	Print	irrigation.
	Condition moisture & rain	Print	
	End	Pay Roll Control	