

## PROGRAMMING FUNDAMENTAL

## ASSIGNMENT

Q-1

Miles per hour to kilometer per  
seconds

## (a) IPO CHART:

Input	Process	Output
mph	Convert mph to km/h	km/s
.	Convert km/h to km/s	

## (b) PSEUDOCODE:

Input:

Speed in miles per hour (mph)

Process:

mph to km/h

$$\text{km/h} = \text{mph} * 1.609$$

Date: \_\_\_\_\_

Day: \_\_\_\_\_

Kilometer per hour to kilometer  
per second

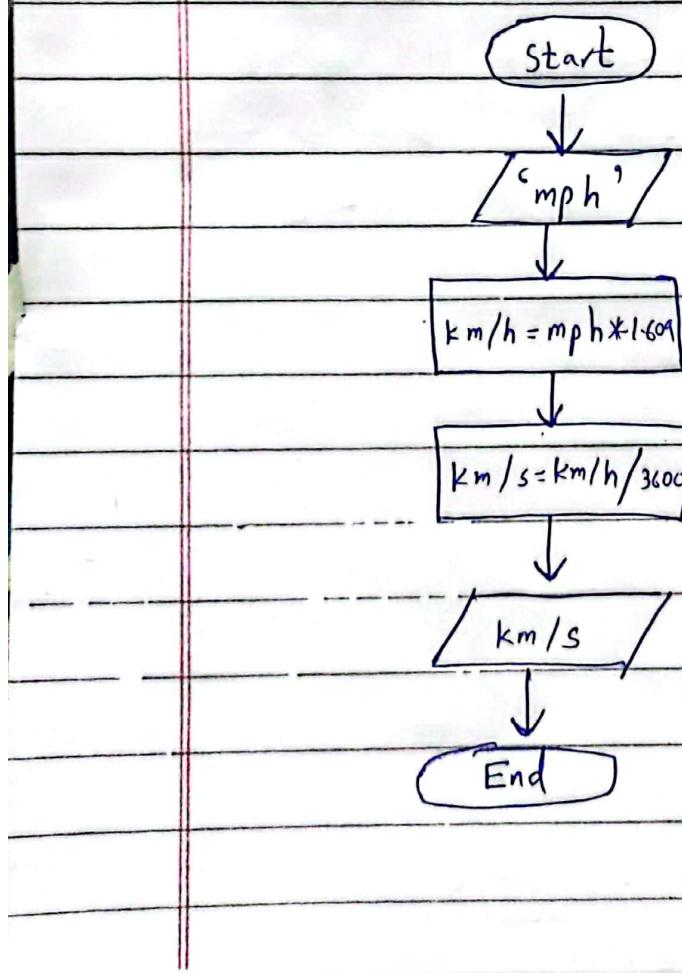
$$\text{km/s} = \text{km/h} / 3600$$

Output:

Speed in kilometer per second  
km/s

END

(c) FLOW CHART



Q-2

## EVEN OR ODD

## (a) IPO CHART

Input	Processing	Output
'n'	Check if n is greater than 0	Even (divisible by 2)
	Determine even or odd using modulus	Odd (not divisible by 2)
	if $n < 0$ then its invalid	Invalid if $n < 0$

## (b) PSEUDOCODE:

Start

Input: 'n'

Process:

if  $n > 0$ then if  $n \% 2 == 0$ 

Print "Even"

Date: \_\_\_\_\_

Day: \_\_\_\_\_

Else

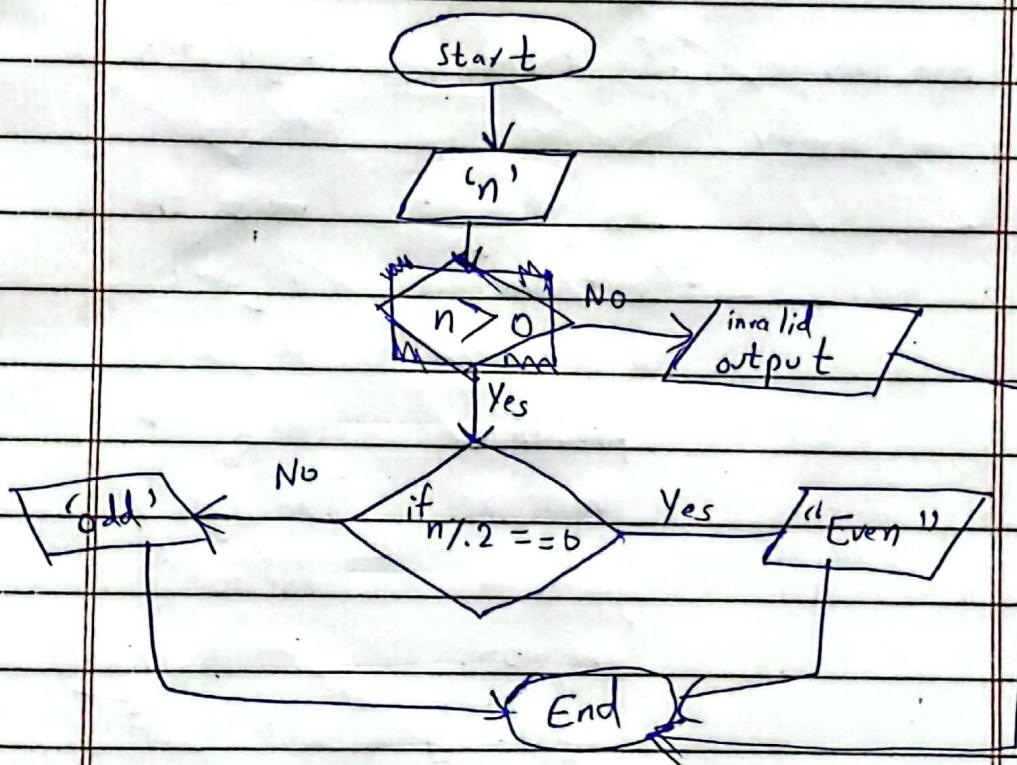
Print "Odd"

If  $n < 0$

then print "Invalid Output."

"END

### (c) Flow Chart:



Q - 3

## LEGAL AGE OF MARRIAGE IN PAKISTAN

### (a) IPO CHART

Input	Processing	Output
Age, Province and Gender	<ul style="list-style-type: none"><li>Check the province for legal age.</li><li>Check the age for both the provinces.</li><li>Check the gender in punjab as well.</li></ul>	Legal if the age meets the requirement of province illegal if the age does not meet the criteria of province.



Date \_\_\_\_\_

Day \_\_\_\_\_

## (b) PSEUDOCODE

Start

Input:

- Age
- Gender
- Province

Process:

If province == Sindh

then if age  $\geq 18$

Print "Legal"

Else

Print "Illegal"

End

If province == "Punjab"

then if gender == "Female"

then if age  $\geq 16$

then Print "Legal"

else print "Illegal"

Else if gender == "Male"

then if age  $\geq 18$

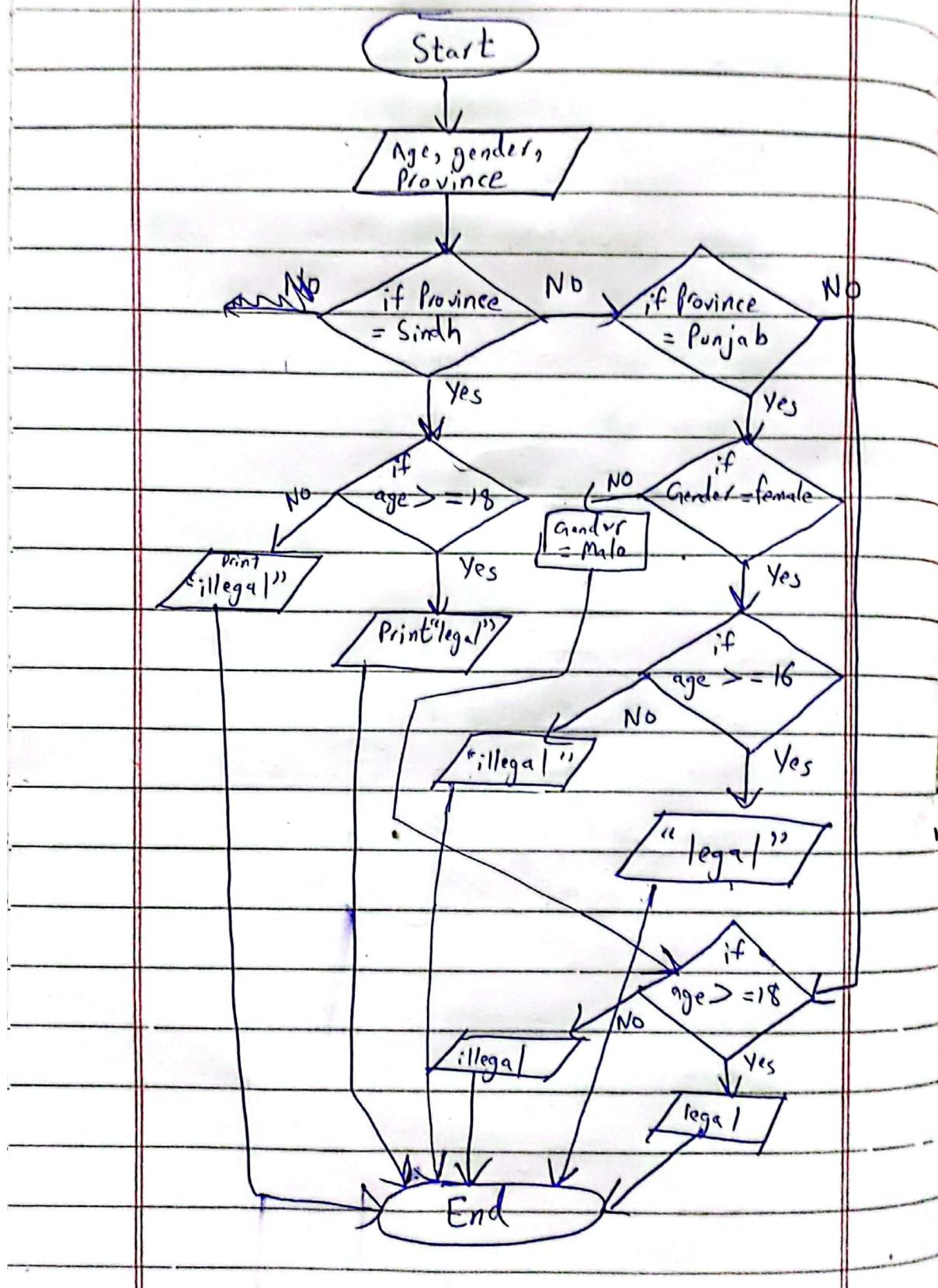
Print "Legal"

Else

Print "Illegal"

End

## (C) FLOW CHART



## Q-4 GROCERY CALCULATOR

### (a) IPO CHART :

Input	Processing	Output
Total budget	If ( $\text{budget} > \text{price}$ )	Price
Price for each vegetable	$\text{Price} = \text{Price of each vegetable}$	Remaining budget
Amount of each vegetable	$x \text{ amount of each vegetable}$	Insufficient amount, decrease
	Remaining amount $= \text{total budget} - \text{Price}$	the amount of vegetable
	Else	
	print "Insufficient amount, decrease the amount of vegetable".	

## (ii) PSEUDOCODE:

Start

- Input • Total budget  
• Price of vegetable  
• Amount of vegetable

Processing :

$$\text{Price} = \text{Amount} \times \text{price of vegetable}$$

If

$$\text{Total budget} > \text{Price}$$

Then

$$\text{Remaining} = \text{Total budget} - \text{Price}$$

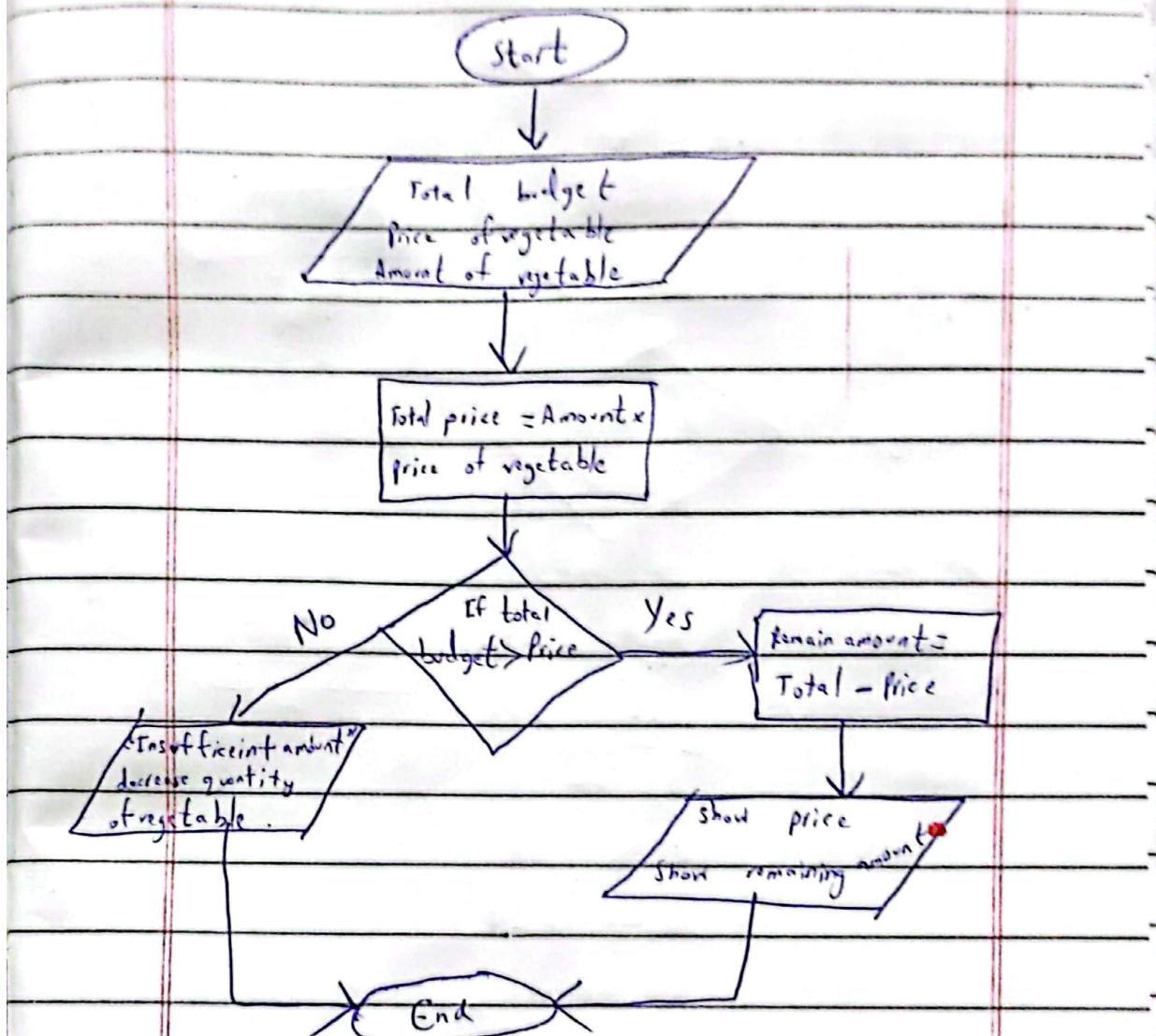
Show Remaining amount and Price

Else

Show "Decrease the amount of vegetable".

End

### 3) FLOW CHART:



**Q - 5**

## CROP MANAGEMENT

### (i) IPO CHART:

Input	Process	Output
Crop type	Read crop type, moisture level	Soil moisture level
Soil moisture level	and rain status Apply irrigation rule	Whether irrigation is needed
Rainfall status	For wheat: If moisture < 30%, and no rain in 24 hour initiate irrigation	
	• For corn moisture < 40% irrigation initiate rain	
	For rice moisture < 25%, with no rain in 24 hour initiate irrigation	

### (iii) PSEUDOCODE:

Start

Ask the user for moisture (in %)

Ask the user time of rain (in hours)

Ask the type of crop from user

Set moisture = m

Set rainfall to = n

If crop = wheat

If  $m < 30$  and  $n \geq 24$

Start irrigation

Else

No irrigation

Then if Type = corn

If  $m < 40$

start irrigation

Else

Don't start irrigation

Then if Type = rice

If  $m < 25$  and  $n \geq 24$

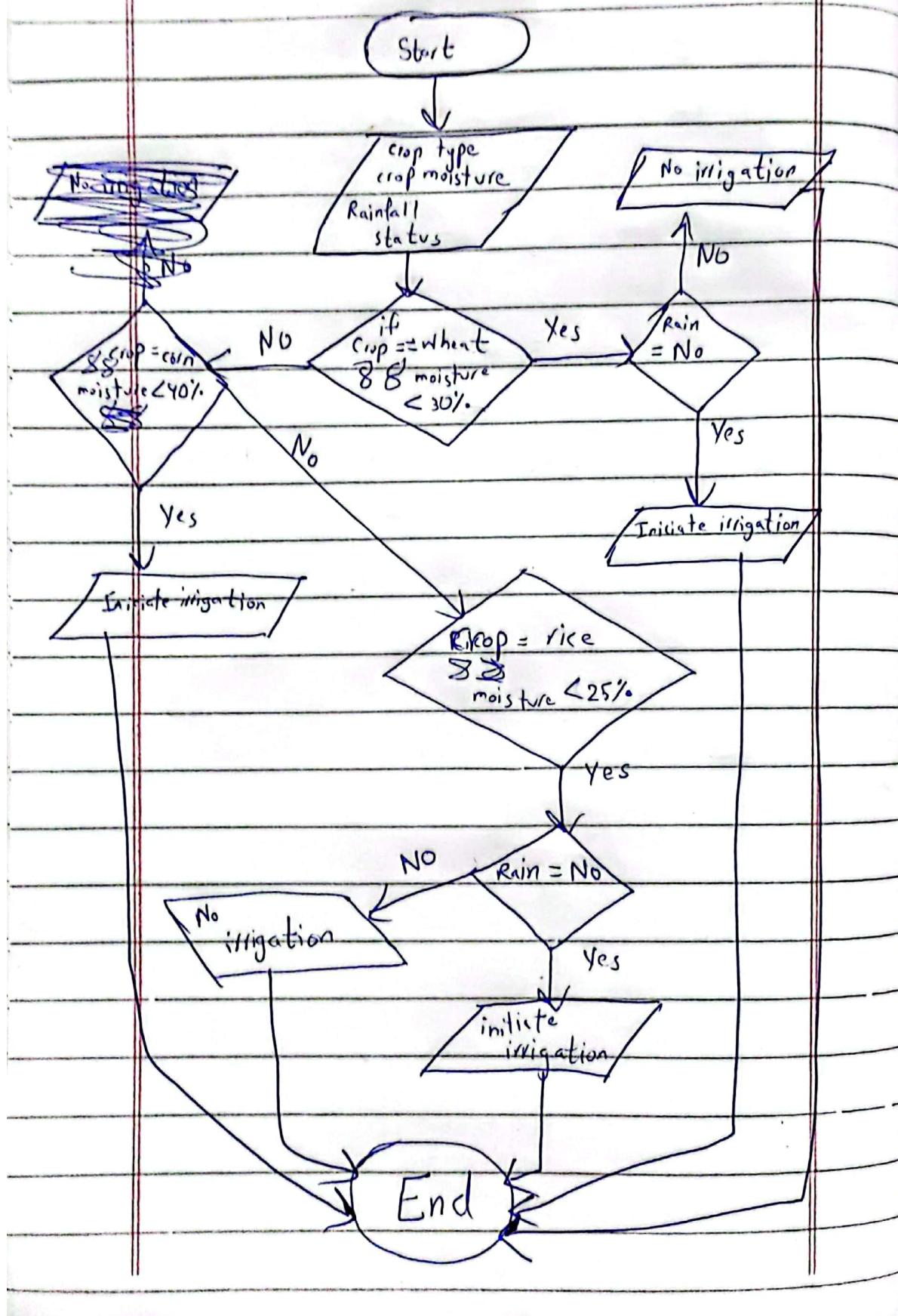
start irrigation

Else (No irrigation)

End if

End

## (iii) FLOW CHART



## Q-6

### RIDE ELIGIBILITY

(i) IPO CHART:

Input	Process	Output
height, age, selected riders	Read age, height and criteria	Eligibility for the ride
	For Dragon	
	Roller coaster	
	Height $\geq 48$ and Age $\geq 10$	
.	For Sky Swing	
.	Height $\geq 54$ .	
	For Carousel	
	Age $\geq 5$	

(ii) PSEUDO CODE:

Start

Input: height, age, ride

Processing:

If ride == Dragon Roller coaster

then if h >= 48 and Age >= 10

Then

Print "Eligible for Dragon Roller  
Coaster"

Else

"Not Eligible"

Else if

Ride = Sky Swing

then if h >= 54

Print "Eligible for Sky Swing"

Else

Print "Not Eligible"

Else if

Ride = Crouse

then if age >= 5 years

Print "Eligible for Crouse"

Else

Print "No eligible"

Else

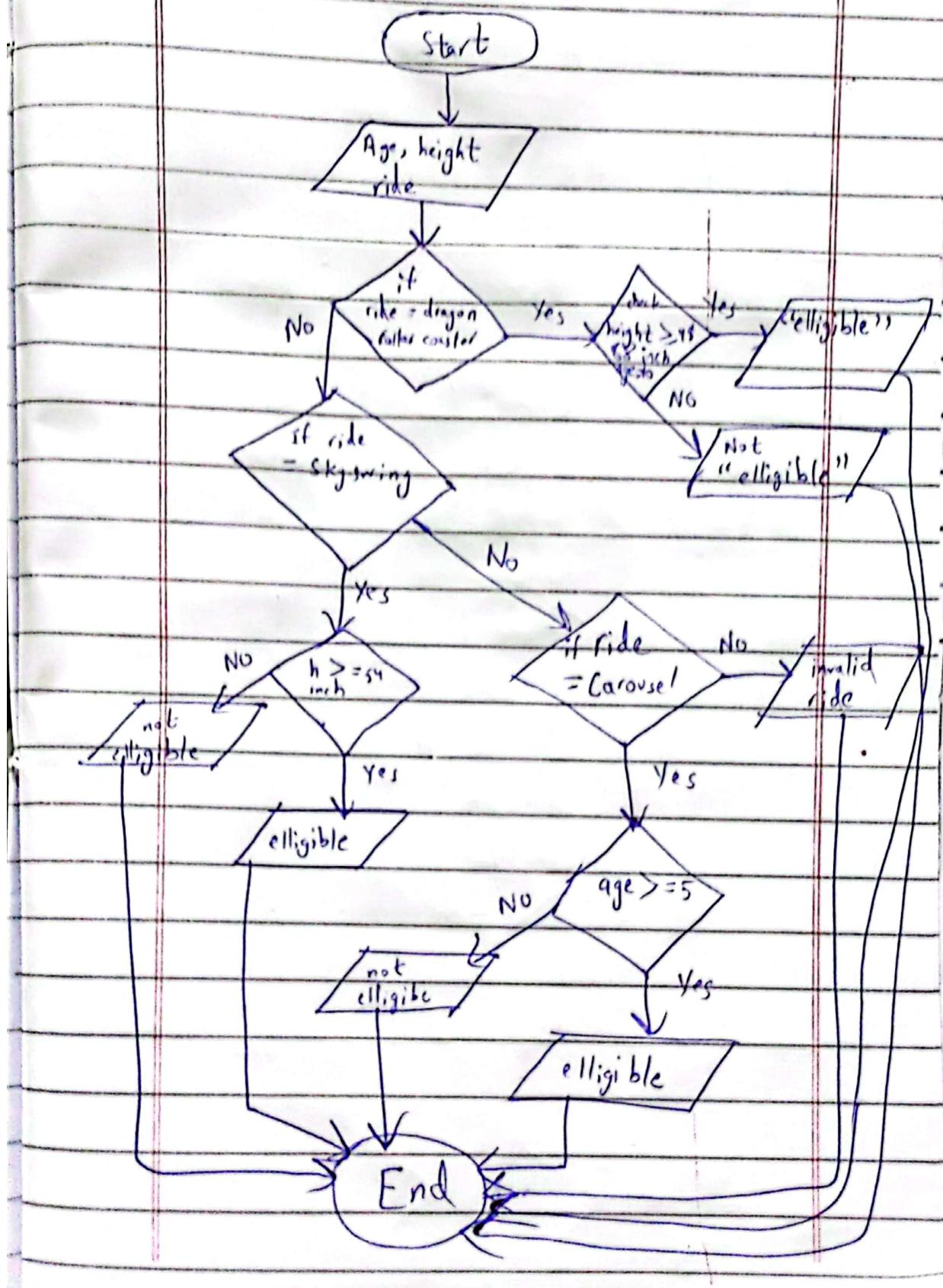
Print "Invalid ride"

End

Date: \_\_\_\_\_

Date: \_\_\_\_\_

## FLOWCHART:



Q - 7

## WHICH FLOOR

### (a) IPO CHART:

Input	Process	Output
Binary Number of 7 digits	Convert 7 digit input into corresponding floor based on the pattern	Floor number (0 to 9)
	Match	
	the binary for each floor	

## (ii) PSEUDOCODE:

Start

Input : Binary numbers

Processing:

If binary = "0110000" Then  
Print "1"

Else if binary = "111110" ten  
Print "0"

Else if binary = "0110010" then  
Print "2"

Else if binary = "1011011" then  
Print "3"

Else if binary = "1100101" then  
Print "4"

Else if binary = "101101" then  
Print "5"

Else if binary = "101111" then  
Print "6"

Else if binary = "0010010" then  
Print "7"

Else if binary = "111111" then  
Print "8"

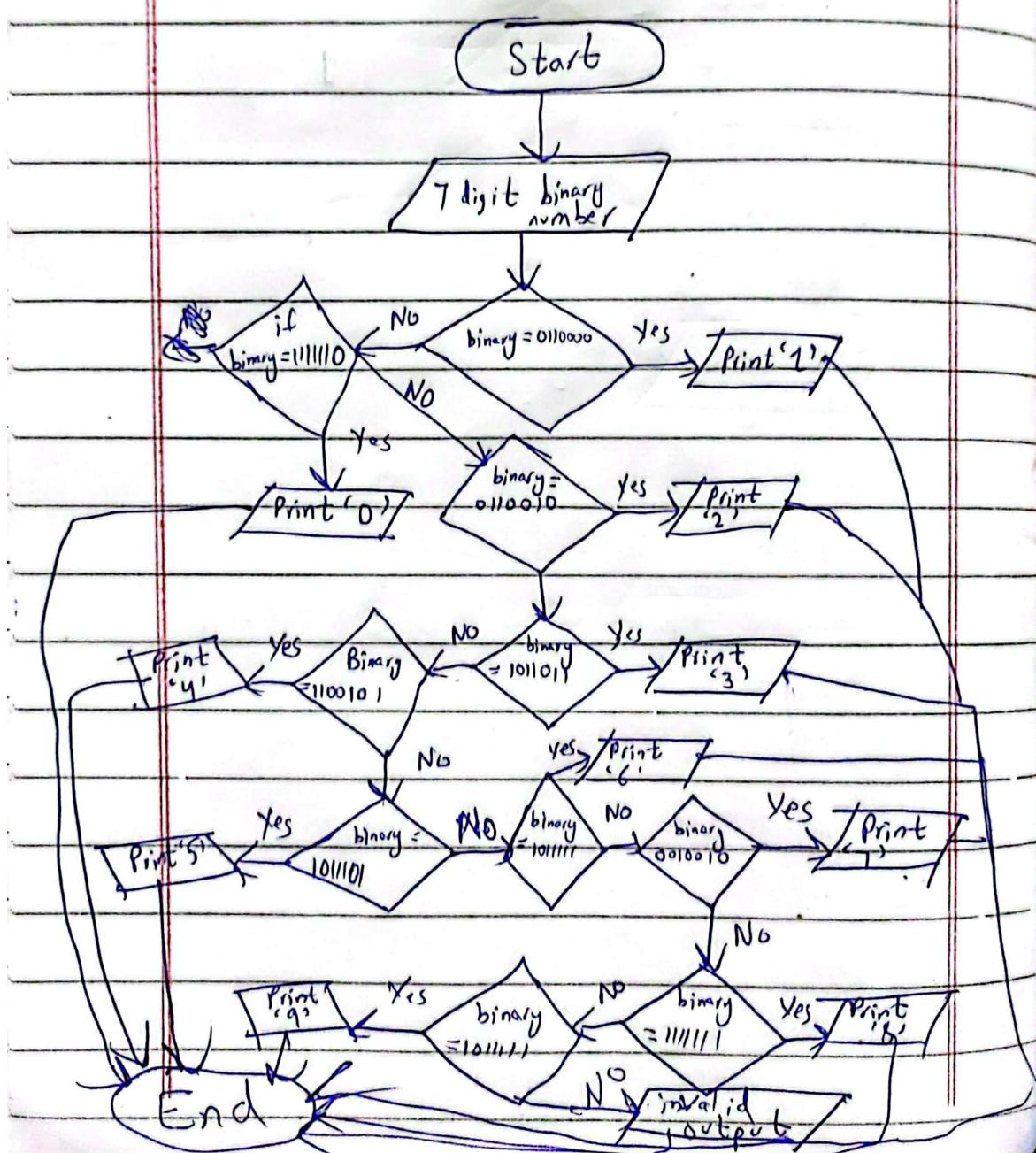
Else if binary = '101111' then  
Print "g"

Else

Print "Invalid input"

End

### (iii) FLOW CHART



**Q - 8**

## DIGIT SUM

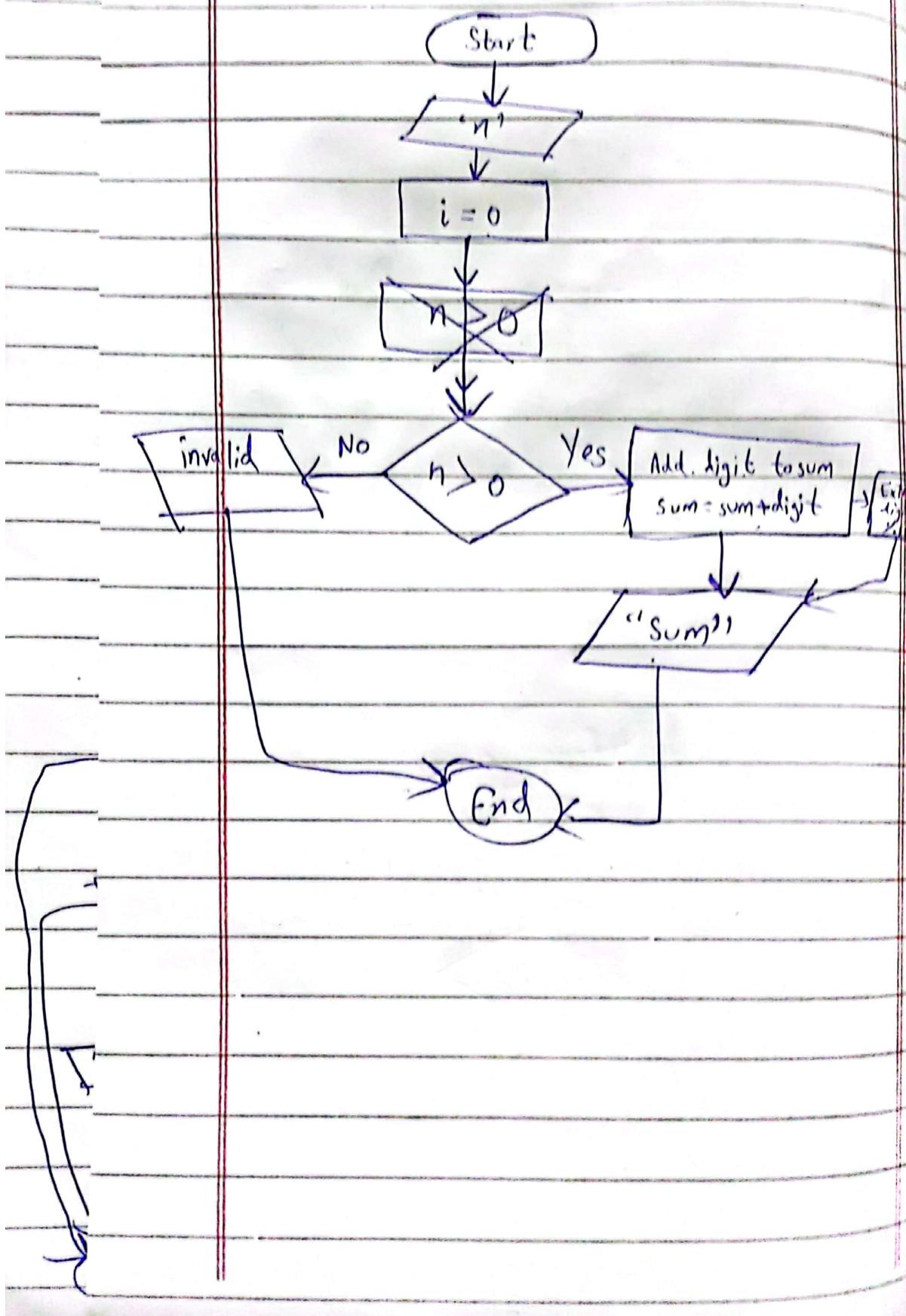
### (i) IPO CHART:

Input	Process	Output
'n'	Extract each digit Sum the extracted digit	Sum of the digits

### (ii) PSEUDOCODE:

- 1) Start
- 2) Ask the user to enter a number
- 3) Add each of digit
- 4) Sum of digit to digit
- 5) Display result
- 6) End

## (iii) FLOW CHART:



Date:

Day:

Q-9

## EXTRACT

### (i) IPO CHART:

Input	Process	Output
DOB Today's date	<ul style="list-style-type: none"><li>Validate DOB and today's date</li><li>Calculate age in years month and days</li><li>Adjust for leap year</li></ul>	<ul style="list-style-type: none"><li>Validate Date</li><li>Age</li></ul>

### (ii) PSEUDOCODE:

1) Start

Input: Ask user to enter a number

Date: \_\_\_\_\_

Day

Ask the user to enter birthday  
in DD/MM/YY format

Today date in DD/MM/YY format

Process:

If day are not same unless it  
February with 1 day

Print Please correct data

Else

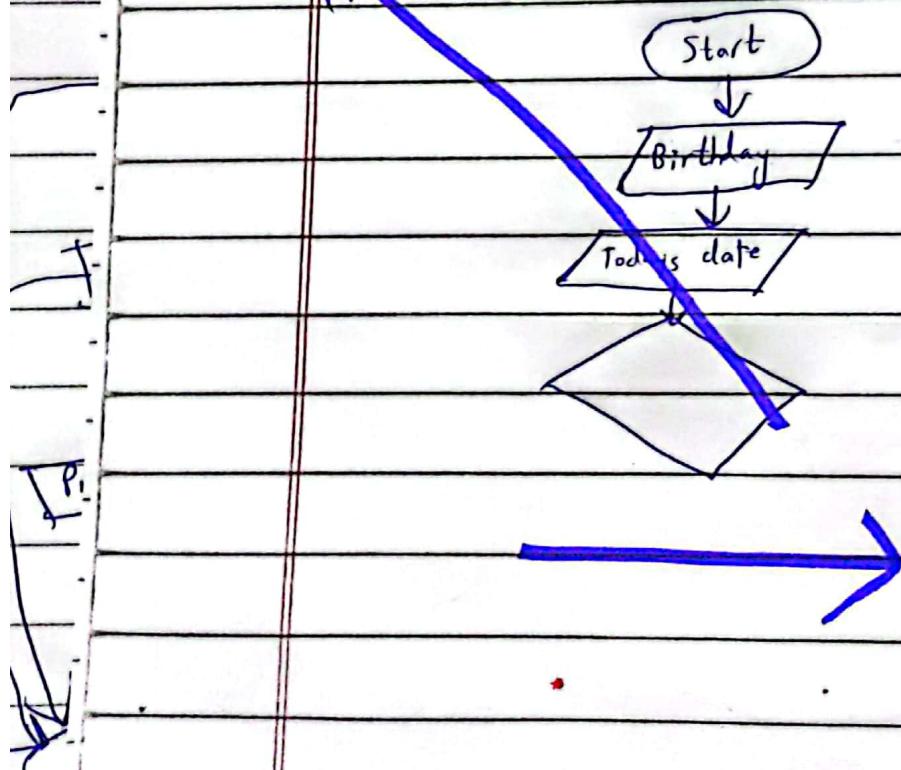
Result = Today's date - birthday date

End if

Display result

End

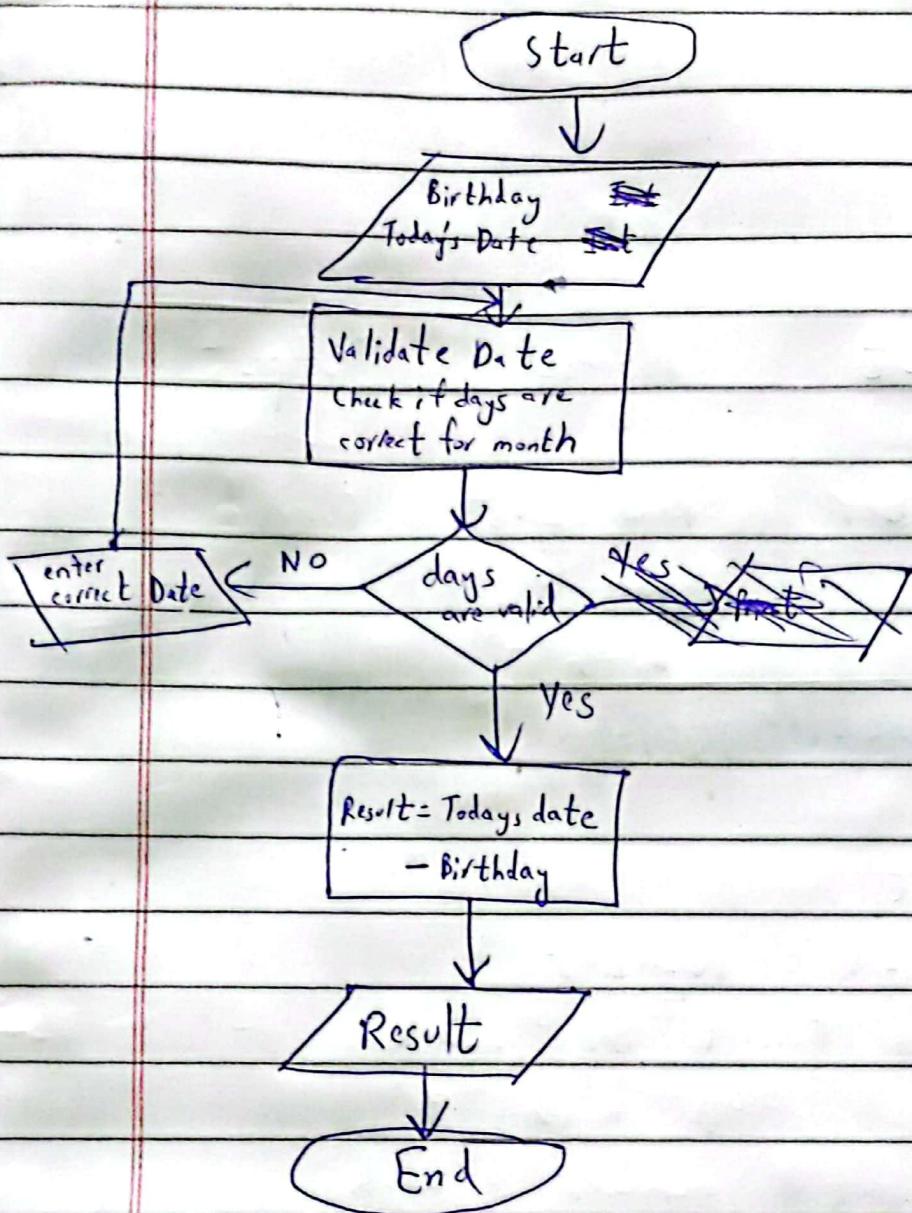
## Flow chart:



Date: \_\_\_\_\_

Day: \_\_\_\_\_

### (iii) FLOW CHART



Q - 10

## FAULTY KEYBOARD

### (i) IPO CHART:

<u>Input</u>	<u>Process</u>	<u>Output</u>
Number	Read Input	Corrected
	Replace "90" with "9"	Number

### (ii) PSEUDOCODE:

Start

Input : Number

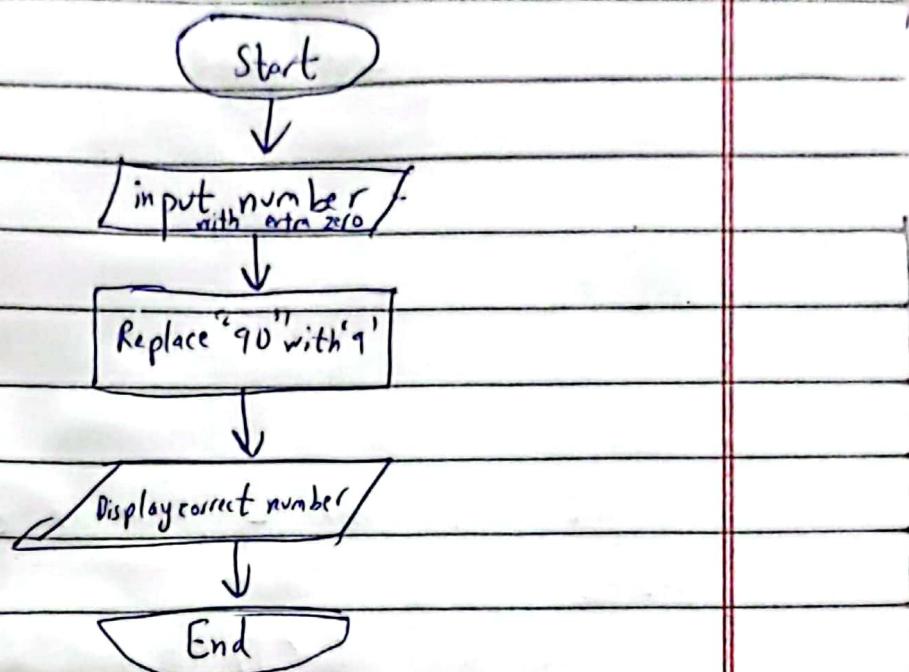
Process: Correct by placing  
the number of  
"90" with "9"

Corrected number = Replacing  
"90" with "9"

Result      Display number

End

## (iii) FLOW CHART:



Q-11

## COPRIME

## (iv) IPO CHART:

Input	Process	Output
Any two Numbers greater than 0	Calculate GCD for two numbers  If GCD is 1,	Whether numbers are coprime or not

Then number are

coprime  
Else Print  
Not

### (iii) PSEUDOCODE

Start

Input: Number 1, Number 2

Process

Calculate G.C.D  $\begin{cases} \text{Temp} = b, b = a \% b \\ a = \text{temp} \end{cases}$

If G.C.D == 1

Then

Print "Number are coprime"

Else

Print "Number are not coprime"

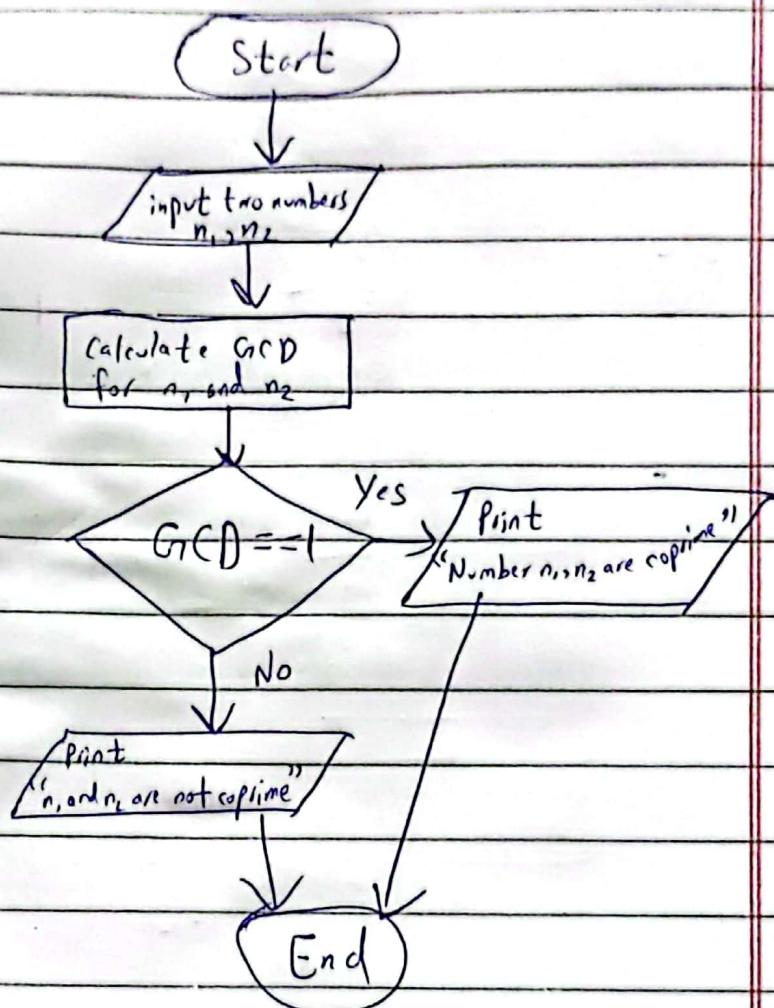
End if

End

### (iii) FLOW CHART



## (iii) FLOW CHART:



Q - 12  
DEFUSE THE BOMB

## (i) IPO CHART:



Date:	Day	
Input	Process	Output
5 liter	Fill 5	
and 3	liter jug	4 litres
Liter	Pour 5 liter	
jug	jug to 3 liter	
	jug	
	Empty 3 liter	
	jug	
	Pour the	
	remaining 2 liter from	
	5 liter to	
	3 liter jug	
	Fill 5L jug	
	again	
	Pour 5L into	
	3L jug	
	until 3L <del>is</del> jug	
	is full	
	The 5L jug will now have	
	4L	

Date: \_\_\_\_\_

Day: \_\_\_\_\_

## PSEUDO CODE:

Start

Input 5 liter and 3 liter jug

Process:

Fill 3<sup>liter</sup> Jug

Fill 5L jug from 3L jug

Fill 3L jug

Fill 5L jug from 3L jug

Empty 5L jug

Fill 5L jug from 3L jug

Fill 3L jug

Fill 5L jug from 3L jug

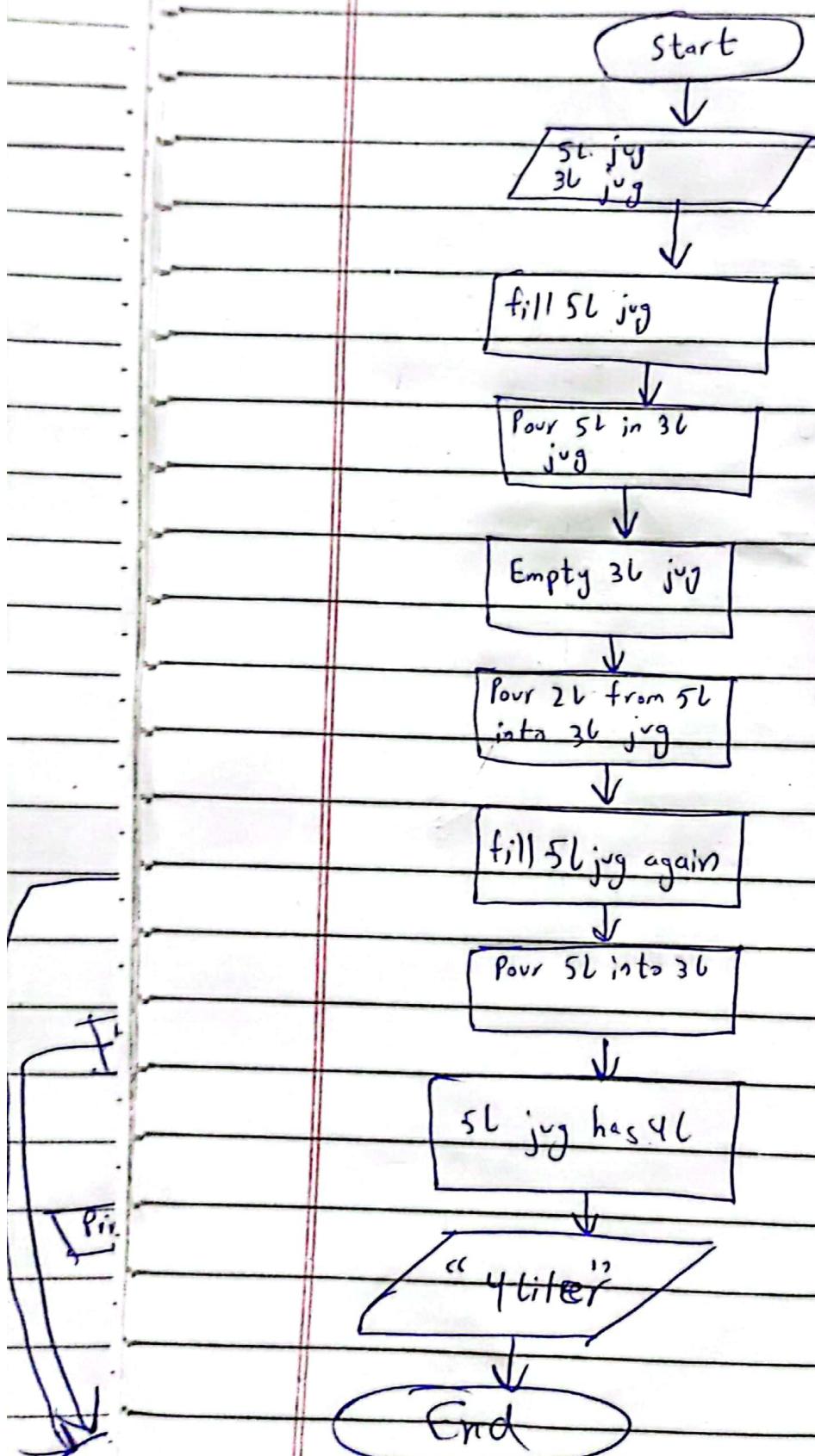
5L jug will have 4L

End



1

## (iii) FLOW CHART:



Q-13

### (i) IPO CHART:

Input	Process	Output
$k, l$	<p>Store value in gcd.</p> <p>Amount of water multiple of gcd of <math>k, l</math> can be measured.</p> <p>All possible quantity of water is measured.</p>	<p>Print "The max amount and all possible quantities that can be measured."</p>
	<p>Next loop calculate all possible water quantity as input</p>	

### (ii) PSEUDO CODE:

Start

Input 'k', 'l'

Process

$$\text{gcd} = k$$

$$a = l$$

While  $a \neq 0$

$$\text{remainder} = a$$

End while

$$\text{max} = k + l - \text{gcd}$$

Print "Max quantity is max"

Print "Possible quantities"

$$i = 1;$$

While ( $i \leq \text{max/gcd}$ )

Print

$$i * \text{gcd}$$

End While

End

Date: \_\_\_\_\_

Day: \_\_\_\_\_

## (iii) FLOW CHART

