**Assignment -1 PF Theory**

**Name: Bharat Roll No: 25K-0165 Section: B**

**Part 1 Question 1:**

1. Analyze using PAC, IC and IPO Charts.

* **IPO Chart (Input-Process-Output)**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Transaction amount, country, time, count | If amount > limit, mark suspicious, | Flag/No flag |
|  | If country != usual and is foreign, mark suspicious | Flag/No flag |
|  | If transactions in hour > limit, flag the extras | Flag/No flag |

**Algorithm:**

1. Start.
2. Input daily amount, transaction country, transaction times/count, user’s regular country.
3. If daily amount > fixed limit, flag transaction as suspicious.
4. If transaction country is foreign and not in user’s usual countries, flag as suspicious.
5. If more than three transactions occur within same hour, flag additional transactions as suspicious.
6. Output results.
7. End.

**Pseudocode:**

START

input amount, transaction \_country, transaction count, usual countries

If amount > daily\_limit Then

PRINT "Transaction flagged: Amount limit exceeded"

ELSE IF transaction\_country NOT IN usual countries THEN

PRINT "Transaction flagged: Unusual country entry"

ELSE IF transaction count > 3 THEN

PRINT "Transaction flagged: Too many transactions in hour"

ELSE

PRINT "Transaction approved"

END.

**PAC (Problem Analysis Chart):**

|  |  |
| --- | --- |
| Given Data | Required results |
| Daily transaction amount, transaction country, time and count of transactions within each hour, user’s usual country pattern. | Flag transactions as suspicious based on criteria**.** |
| Process | Alternative solutions |
| Compare amount with limit, check country, count hourly transactions. | * Manually transactions by bank staff. |

**IC (Interactivity Chart):**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Transaction amount, transaction country, time of transaction | Decision steps for flagging suspicious behavior | Suspicious transaction indication or approval |

**Question 2:**

1. Analyze using PAC, IC and IPO Charts.

* **IPO Chart (Input-Process-Output)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input** | | | **Process** | | **Output** |
| Values of x, y. | | | If x > 0 and y > 0 → Quadrant I | | Quadrant I message |
|  | | | If x < 0 and y > 0 → Quadrant II | | Quadrant II message |
|  | | | If x < 0 and y < 0 → Quadrant III | | Quadrant III message |
|  | | If x > 0 and y < 0 → Quadrant IV | | | Quadrant IV message |
|  | If x==0 or y==0 → On axis | | | Axis messege | |

**Algorithm:**

1. Start
2. Input x and y coordinates
3. If x > 0 and y > 0, print "Quadrant I"
4. If x < 0 and y > 0, print "Quadrant II"
5. If x < 0 and y < 0, print "Quadrant III"
6. If x > 0 and y < 0, print "Quadrant IV"
7. If x == 0 or y == 0, print "Axis"
8. End

**Pseudocode:**

Start

Input x, y

IF x > 0 AND y > 0 Then

PRINT "The point (x, y) lies in quadrant I"

ELSE IF x < 0 AND y > 0 THEN

PRINT "The point (x, y) lies in quadrant II"

ELSE IF x < 0 AND y < 0 THEN

PRINT "The point (x, y) lies in quadrant III"

ELSE IF x > 0 AND y < 0 THEN

PRINT "The point (x, y) lies in quadrant IV"

ELSE

PRINT "The point (x, y) lies on an axis"

END.

**IC (Interactivity Chart):**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Obtain values of x and y coordinates. | Evaluate x and y, determine and display quadrant. | Display quadrant result. |

**PAC Chart:**

|  |  |
| --- | --- |
| **Given Data** | **Required results** |
| Input x and y coordinates of a point. | Quadrant where the point lies. |
| **Process** | **Alternative solutions** |
| Check the signs of x and y. | Use decision structures to determine the quadrant. |

**Question 3:**

1. Analyze using PAC, IC and IPO Charts.

* **IPO Chart (Input-Process-Output)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input** | | | **Process** | | **Output** |
| Age, eyesight test, written test, driving test, certificate | | | If age < 18 → Ineligible; else check each step with IF statements as below | | Eligible/Ineligible |
|  | | | Eyesight passed → continue; failed → needs glasses | |  |
|  | | | Written test passed → continue; failed → ineligible | |  |
|  | | Driving test passed → continue; failed → ineligible | | |  |
|  | Driving test passed → continue; failed → ineligible | | |  | |
|  | If age > 60, check medical certificate; if not, eligible | | |  | |

**Algorithm:**

1. Start
2. Input age. If age < 18, output “Not eligible”
3. Input eyesight test (P/F). If F, output “Needs prescription glasses”; if P, continue
4. Input written test (P/F). If F, output “Not eligible”; if P, continue
5. Input driving test (P/F). If F, output “Not eligible”; if P, continue
6. If age > 60, input medical fitness certificate (Y/N). If N, output “Not eligible”; if Y, output “Eligible”
7. If age ≤ 60 and all previous steps are P, output “Eligible”
8. End

**Pseudocode:**

START

INPUT age

IF age < 18 THEN

PRINT "Ineligible for license"

ELSE

INPUT eyesight\_test

IF eyesight\_test == 'F' THEN

PRINT "Needs prescription glasses"

ELSE

INPUT written\_test

IF written\_test == 'F' THEN

PRINT "Ineligible for license"

ELSE

INPUT driving\_test

IF driving\_test == 'F' THEN

PRINT "Ineligible for license"

ELSE

IF age > 60 THEN

INPUT medical\_certificate

IF medical\_certificate == 'Y' THEN

PRINT "Eligible for license"

ELSE

PRINT "Ineligible for license"

ELSE

PRINT "Eligible for license"

END.

**Question 4:**

1. Analyze using PAC, IC and IPO Charts.

* **IPO Chart (Input-Process-Output)**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Five integers (1-13) | Count each rank’s frequency | “Full house” or “Not...” |
|  | Check for three of one, two of another. |  |

**Algorithm:**

1. Start
2. Input five card ranks
3. Count frequency of each rank (max two different ranks)
4. If there are exactly two distinct ranks:
   * One occurs three times
   * One occurs two times  
     Output “Full house”
5. Else output “Not a full house”
6. End

**IC (Input-Process-Output Chart):**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Five card ranks (integers between 1 and 13) | Count the frequency of each rank; check for 3-of-a-kind and 2-of-a-kind. | Message indicating whether hand is a full house |
|  |  |  |

**PAC Chart:**

|  |  |
| --- | --- |
| **Given Data** | **Required results** |
| Five integers (card ranks) in the range 1–13 representing the poker hand. | State whether the hand is a full house. |
| **Process** | **Alternative solutions** |
| Count the occurrences of each card rank; a full house must contain three of one rank and two of another | Frequency counting method. |

**Pseudocode:**

START

INPUT cards[5]

SET freq1 = 0, freq2 = 0

SET value1 = cards[0]

SET value2 = -1

# Identify distinct values

FOR i = 1 to 4 DO

IF cards[i] != value1 THEN

value2 = cards[i]

BREAK

// Count frequencies

FOR i = 0 to 4 DO

IF cards[i] == value1 THEN

freq1 = freq1 + 1

ELSE

freq2 = freq2 + 1

// Check for full house

IF (freq1 == 3 AND freq2 == 2) OR (freq1 == 2 AND freq2 == 3) THEN

PRINT "Full house"

ELSE

PRINT "Not a full house"

END

**Question 5:**

1. Analyze using PAC, IC and IPO Charts.

* **IPO Chart (Input-Process-Output)**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| User-entered numbers | If digit (0–9): increment count; else: stop input | Table showing number and occurrences |
|  |  |  |

**Algorithm:**

1. Start
2. Initialize count array for digits 0–9 = 0
3. Repeat:  
   a. Input number  
   b. If 0 ≤ number ≤ 9, increment corresponding count  
   c. Else, stop input
4. Display the table: Number | Number of Occurrences
5. End

**Pseudocode:**

START

For i = 0 to 9: count[i] = 0

REPEAT

INPUT num

IF num >= 0 AND num <= 9 THEN

count[num] = count[num] + 1

ELSE

EXIT LOOP

UNTIL input ends

FOR i = 0 to 9:

PRINT i, count[i]

END.

**IC (Interactivity Chart):**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| User inputs numbers (one at a time) | Check validity (is it 0–9?), record count, terminate on invalid entry. | Display a table of occurrences for each digit. |
|  |  |  |

**PAC Chart:**

|  |  |
| --- | --- |
| **Given Data** | **Required results** |
| Sequence of user-entered numbers (digits 0–9). | Count and display how many times each digit (0–9) is entered. |
| **Process** | **Alternative solutions** |
| Track frequency of each digit, end input if a non-digit is entered. | Use Switch case |

**Question 6:**

**Option (a):** i is 3, j is 5, k is 7

* i<j*i*<*j* (3 < 5): True
  + j<k*j*<*k* (5 < 7): True ⇒ i=j*i*=*j* ⇒ i=5*i*=5
* Print: 5 5 7

**Option (b):** i is 3, j is 7, k is 5

* i<j*i*<*j* (3 < 7): True
  + j<k*j*<*k* (7 < 5): False ⇒ j=k*j*=*k* ⇒ j=5*j*=5
* Print: 3 5 5

**Option (c**): i is 5, j is 3, k is 7

* i<j*i*<*j* (5 < 3): False
  + j>k*j*>*k* (3 > 7): False ⇒ i=k*i*=*k* ⇒ i=7*i*=7
* Print: 7 3 7

**Option (d):** i is 5, j is 7, k is 3

* i<j*i*<*j* (5 < 7): True
  + j<k*j*<*k* (7 < 3): False ⇒ j=k*j*=*k* ⇒ j=3*j*=3
* Print: 5 3 3

**Option (e):** i is 7, j is 3, k is 5

* i<j*i*<*j* (7 < 3): False
  + j>k*j*>*k* (3 > 5): False ⇒ i=k*i*=*k* ⇒ i=5*i*=5
* Print: 5 3 5

**Option (f):** i is 7, j is 5, k is 3

* i<j*i*<*j* (7 < 5): False
  + j>k*j*>*k* (5 > 3): True ⇒ j=i*j*=*i* ⇒ j=7*j*=7
* Print: 7 7 3

**Final Outputs Table**

|  |  |  |
| --- | --- | --- |
| **Case** | **Initial (i,j,k)** | **Printed Output (i,j,k)** |
| (a) | 3,5,7 | 5, 5, 7 |
| (b) | 3,7,5 | 3, 5, 5 |
| (c) | 5,3,7 | 7, 3, 7 |
| (d) | 5,7,3 | 5, 3, 3 |
| € | 7,3,5, | 5, 3, 5 |
| (f) | 7,5,3 | 7, 7, 3 |

**Question: 7**

**Here is the corrected code version.**

#include <stdio.h>

int main() {

int value;

printf("Please enter a value in the range 1...7: ");

scanf("%d", &value);

switch (value) {

case 1:

printf("You entered Monday\n");

break;

case 2:

printf("You entered Tuesday\n");

break;

case 3:

printf("You entered Wednesday\n");

break;

case 4:

printf("You entered Thursday\n");

break;

case 5:

printf("You entered Friday\n");

break;

case 6:

printf("You entered Saturday\n");

break;

case 7:

printf("You entered Sunday\n");

break;

default:

printf("You entered a value out of range!\n");

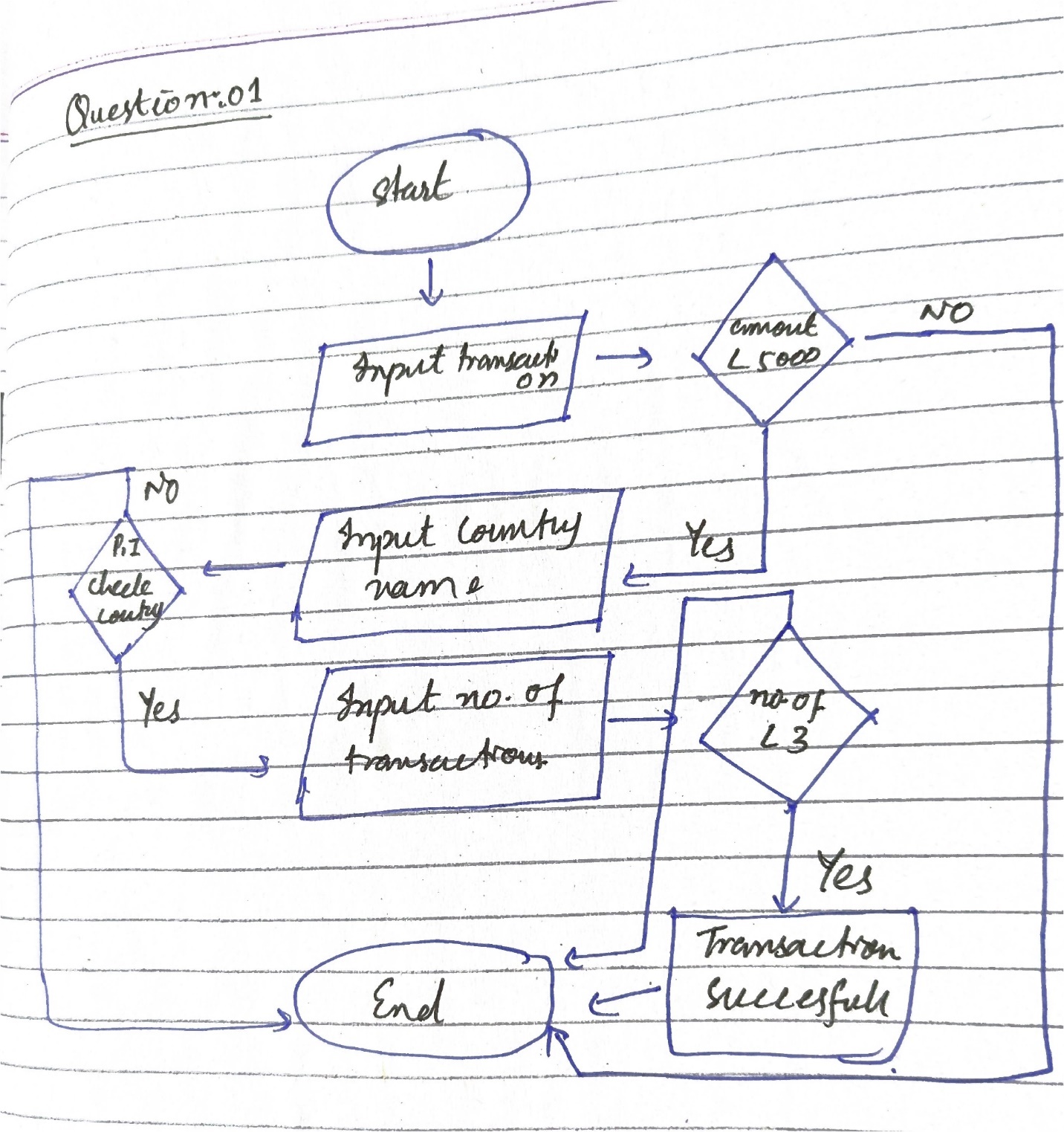
}

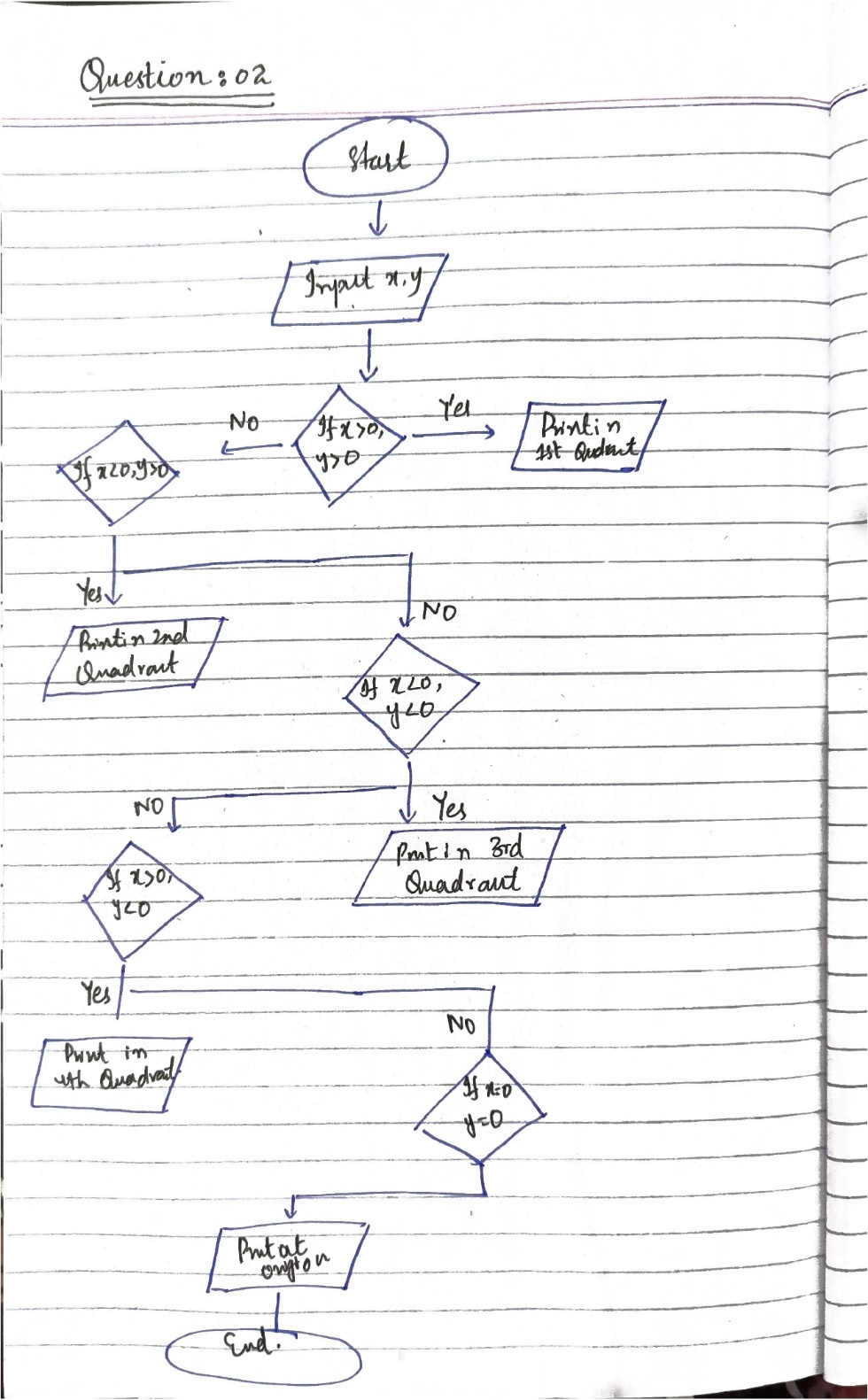
return 0;

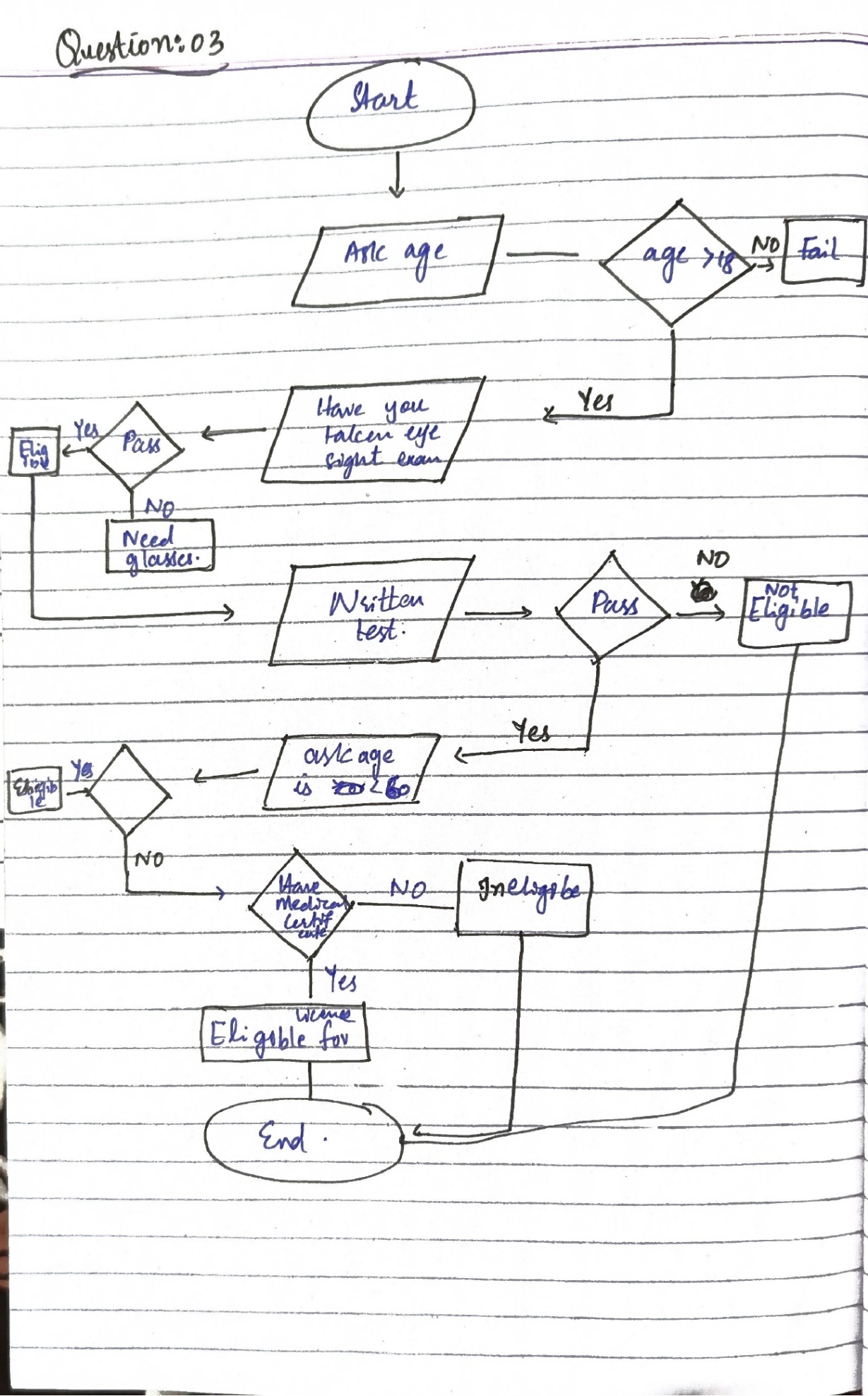
}

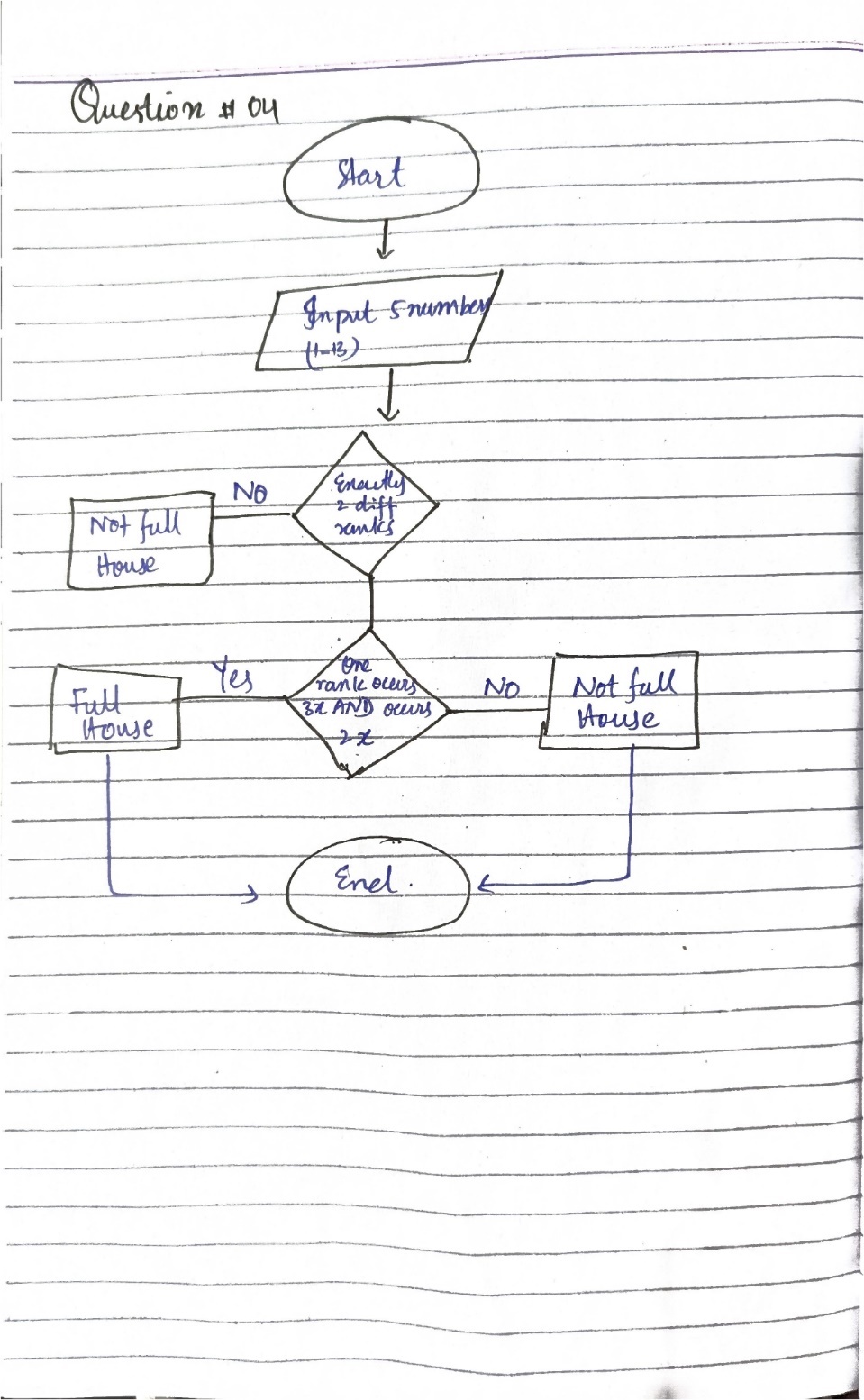
**Flowcharts of All above Questions.**

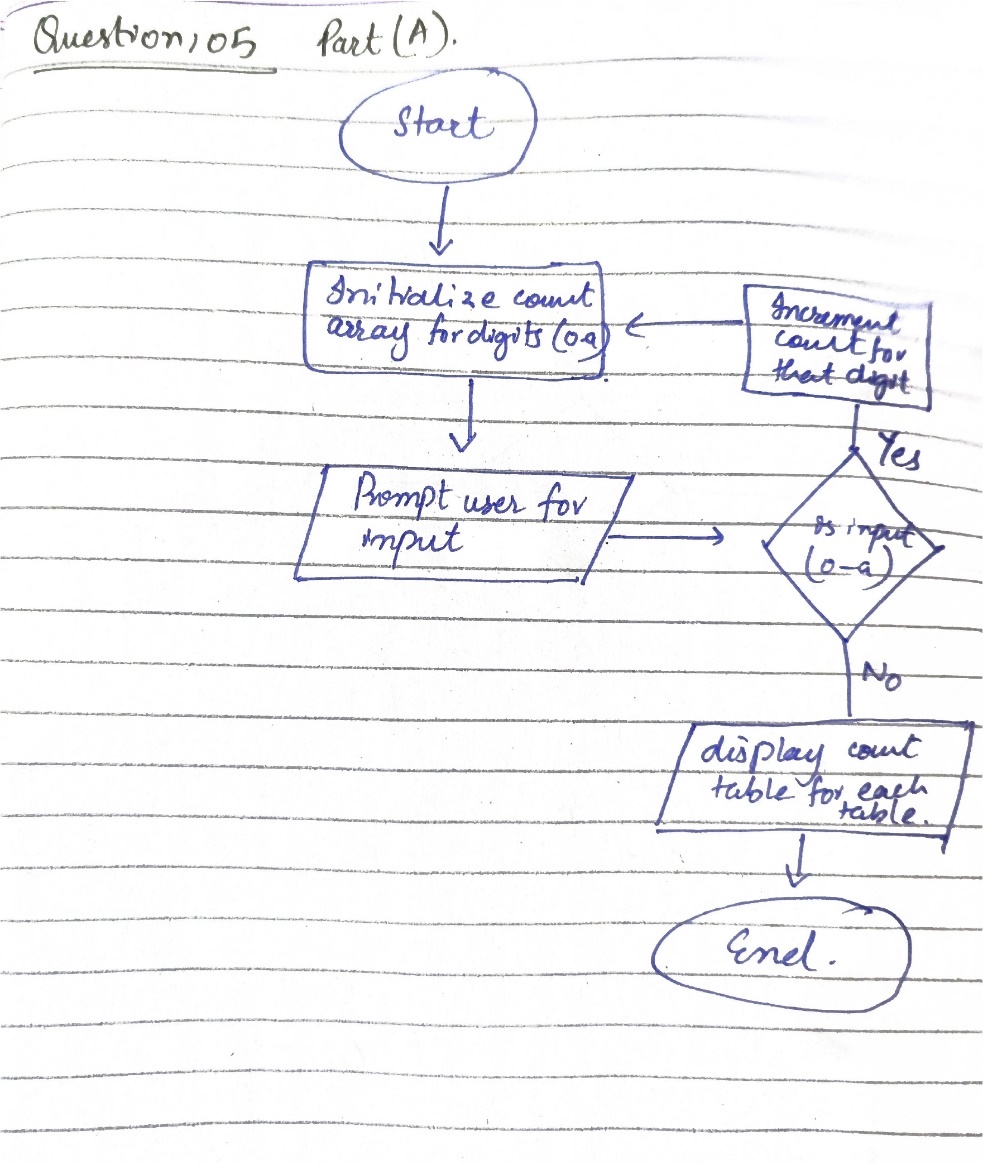
**Question:01**

****

**Question:02**

****

** Question:04**

**Question: 05**

**Question: 03**

**PAC Chart:**

|  |  |
| --- | --- |
| **Given Data** | **Required results** |
| Age, eyesight test result, written test result, driving test result, medical fitness certificate | Inform if applicant is eligible for driving license. |
| **Process** | **Alternative solutions** |
| Sequential decision checks per eligibility rules. | By physically check in balance |

**IC (Interactivity Chart).**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| * User age, results for eyesight, written, driving test, and medical certificate. | * Stepwise decision checks for each criterion. | Message stating eligibility or ineligibility. |
|  |  |  |