# ASSIGNMENT 1 PF THEORY

**NOTE**: C Code has been uploaded on Github and can be access through this link:

(FLOWCHARTS are uploaded In separate file as PNG INCASE THEY ARE NOT VISIBLE IN THIS DOCUMENT)

Q1: PAC Chart

|  |  |
| --- | --- |
| **GIVEN INPUTS**  - Daily spending limit  - Location of Transaction  - Number of transactions in an hour  -an hour is constant(60 minutes) | **REQUIRED OUTPUT**  - Flag transactions that exceed daily limit.  - Uncommon transaction location should be marked suspicious.  - More than 3 transactions in an hour should be marked as suspicious |
| **PROCESSING REQUIRED**  - Customer transaction(s) are detected by bank system.  - If transaction amount > daily limit, print "Flagged".  - If transaction location ≠ customer's common location, print "Flagged".  - If no. of transaction > 3 in an hour, of a customer, print "Flagged" along with each extra transaction. | **ALTERNATIVE SOLUTIONS**  - Spending limit and no. of transactions in an hour can be changed.  - Instead of flagging and marking, the transactions can be blocked to prevent fraudulent activity. |

IPO CHART

|  |  |  |
| --- | --- | --- |
| **INPUT**  - Transaction instance and details  - Daily spending limit  - Location of transaction  - Number of transactions in an hour  An hour/60 min is constant | **PROCESSING**  Read transaction details.  - Read its frequency.  - If amount > daily spending limit, Display "Flagged" with transaction.  - If location is different, Display "Flagged".  - If transaction > 3 in an hour, Display "Flagged". | **OUTPUT**  - Transactions with "Flagged" label, which can be processed by a bank's fraud team. |

**.ALGORITHM**

1.The system takes the input of customer transaction and its frequency.

2. The system checks whether the transaction is less than or equal to the daily spending limit.

3.If the amount exceeds the daily limit, display "Flagged" with the transaction as the system's output.

4. Next, the system checks whether the transaction location is consistent throughout. If a new location is detected for a particular customer, display "Flagged" with the transaction as the system's output.

5. Next, the system checks the number of transactions; or the instance of transactions in an hour(60 min, constant) if the instance exceeds 3, the system displays "Flagged" with the transaction as the system's output.

**.PSEUDOCODE**

START

DECLARE transactionAmt

DECLARE customerName

DECLARE transactionLocation

INIT transactionNum = 0

INIT dailyLimit = $10000

INIT defaultLocation = Pakistan

INIT NumOfTransactionLimit = 3

INIT minutes = 60

WHILE TRUE

INPUT customerName

INPUT transactionAmt

INPUT transactionLocation

transactionNum++

IF transactionAmt > dailyLimit OR transactionLocation != defaultLocation THEN

DISPLAY "FLAGGED"

ELSE

DISPLAY "TRANSACTION IS CLEAR"

ENDIF

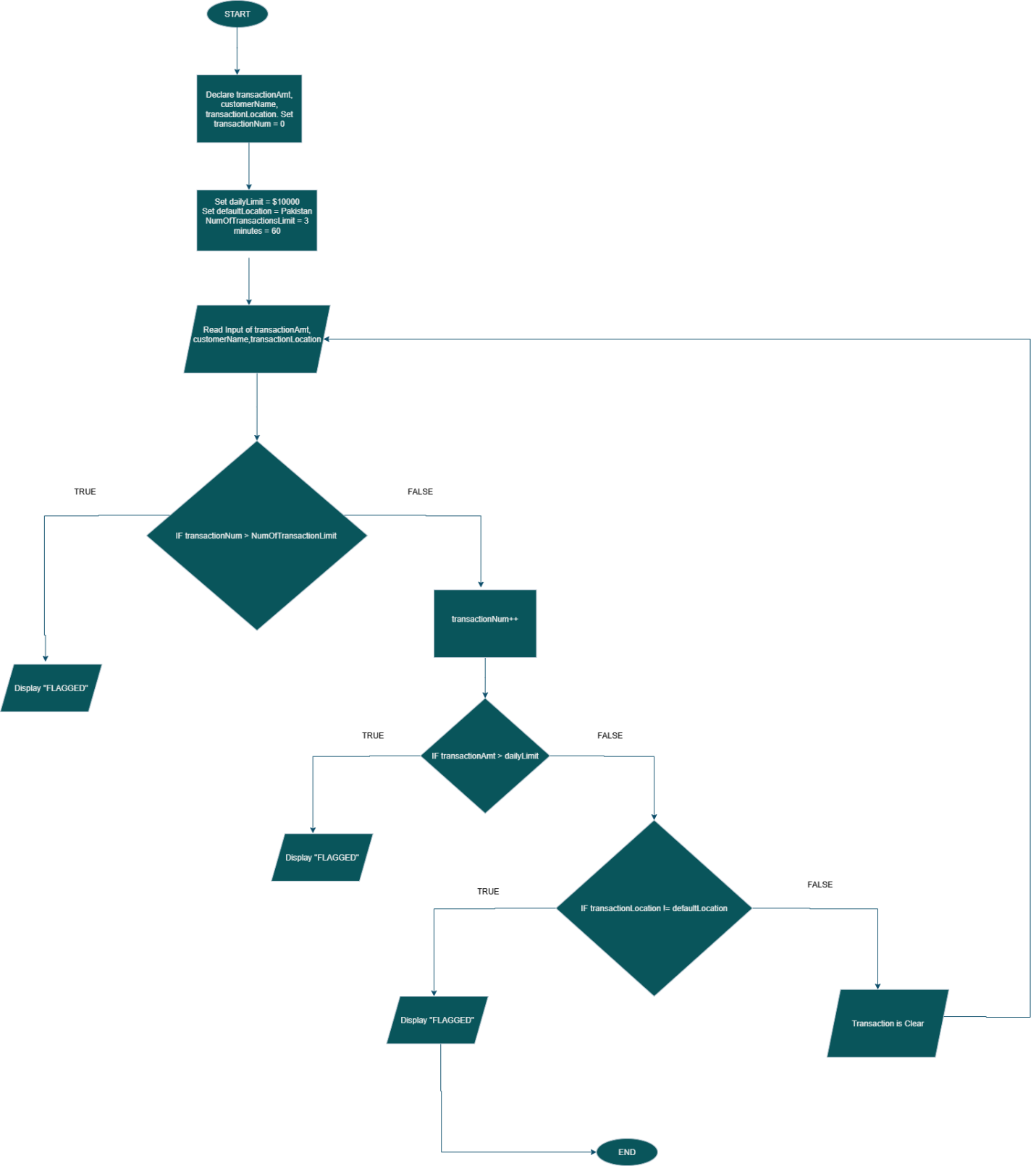
IF transactionNum > NumOfTransactionLimit THEN

DISPLAY "FLAGGED"

ENDWHILE

END

**FLOWCHART**

****

.

.

.

|  |  |
| --- | --- |
| **GIVEN INPUTS**  Quadrant I (+x, +y)  Quadrant 2 (-x, +y)  Quadrant 3 (-x, -y)  Quadrant 4 (+x, -y) | **REQUIRED OUTPUT**  Tell which quadrant do the coordinates lie in  Display “The point (-1, 2) lies in quadrant II” |
| **PROCESSING REQUIRED**  -If x > 0 and y > 0, coordinates lie in Quadrant 1  -If x < 0 and y > 0, coordinates lie in Quadrant 2  -If x < 0 and y < 0, coordinates lie in Quadrant 3.  -If x > 0 and y < 0, coordinates lie in Quadrant 4.  -If x = 0 and y = 0, coordinates lie in origin | **ALTERNATIVE SOLUTIONS**  Using nested conditions, such as  If x > 0, then check whether y < 0 or not, if it is then coordinates like Quadrant 4. |

Q2 PAC CHART

IPO Chart

|  |  |  |
| --- | --- | --- |
| **INPUT**  Input of coordinates  Quadrant I (+x, +y)  Quadrant 2 (-x, +y)  Quadrant 3 (-x, -y)  Quadrant 4 (+x, -y) | **PROCESSING**  -If x > 0 and y > 0, coordinates lie in Quadrant 1  -If x < 0 and y > 0, coordinates lie in Quadrant 2  -If x < 0 and y < 0, coordinates lie in Quadrant 3.  -If x > 0 and y < 0, coordinates lie in Quadrant 4.  -If x = 0 and y = 0, coordinates lie in origin | **OUTPUT**  Display “The point (-1, 2) lies in quadrant II” |

ALGORITHM

1.Ask the user to input coordinates of x-axis and y-axis.

2.If x and y values both are positive, then tell the user the point lies in Quadrant 1.

3.If x is positive and y is negative, tell the user the point lies in Quadrant 4.

4.If x is negative and y is positive, tell the user that point lies in Quadrant 2.

5.If x is negative and y is negative, tell the user that point lies in Quadrant 3.

6.If x and y both are zero, then tell the user that point lies in origin

**PSEUDOCODE**

START

DECLARE x

DECLARE y

REPEAT

DISPLAY “Enter valid x and y coordinates”

INPUT x

INPUT y

UNTIL

IS\_INTEGER(x) AND IS\_INTEGER(y)

IF x = 0 AND y = 0 THEN

DISPLAY “The point lies in origin”

ELSE IF x > 0 AND y > 0 THEN

DISPLAY “The point lies in Quadrant 1.”

ELSE IF x > 0 AND y < 0 THEN

DISPLAY “The point lies in Quadrant 4”

ELSE IF x < 0 AND y < 0 THEN

DISPLAY “The point lies in Quadrant 3”

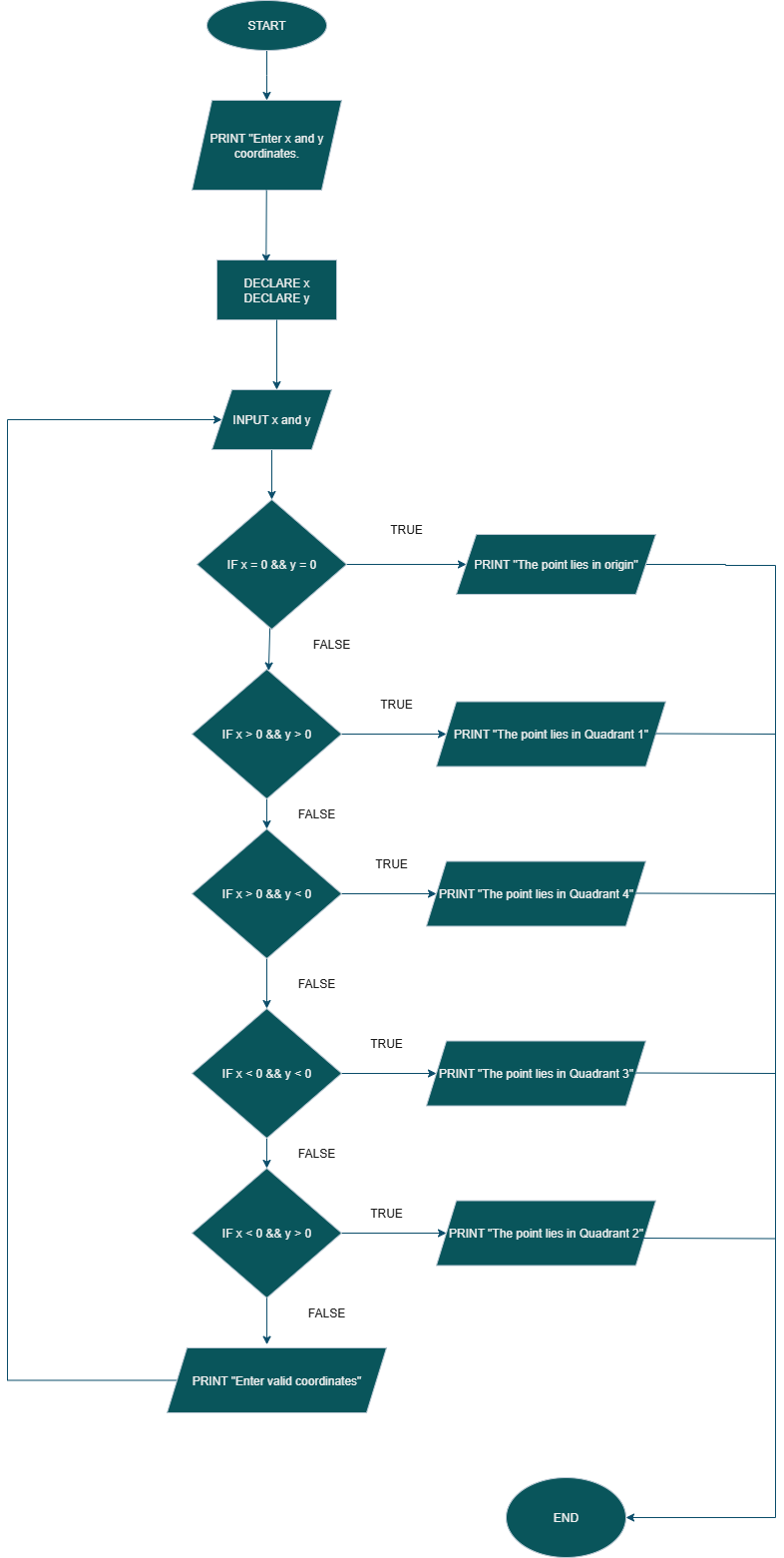
ELSE IF x < 0 AND y > 0 THEN

DISPLAY “The point lies in Quadrant 2.”

ENDIF

END

**FLOWCHART**

****

Q3.PAC CHART

|  |  |
| --- | --- |
| **GIVEN INPUTS**  User Input:  P = Passed, F = Failed  Age  eyesight exam(P/F)  written test(P/F)  Driving Test(P/F)  Medical fitness certificate available or not  For questions regarding passing/failure, only values acceptable are P and F. | **REQUIRED OUTPUT**  Is the user eligible for driving license or not |
| **PROCESSING REQUIRED**  -Validate age.  -Check whether the user has passed the required tests: eyesight, written test and driving test.  -Driving test should be checked with along with age(> 60 or <- 60)  For above 60 users, medical fitness certificate is required,  If at any point, the user does not meet the age requirement or does not pass the test, the next question will not be asked.  If the driving test was failed, or age not validated, the message of ineligibility will be printed  -If medical fitness certificate was not available for age > 60, the message of ineligibility will be displayed  -Ask if they want to recheck | **ALTERNATIVE SOLUTIONS**  Instead of nested conditions, all questions can be asked in one go, and then if they are ineligible, Display in the end that they are ineligible and state the reason e.g “You need a retake of written test as you did not pass it”. |

IPO CHART

|  |  |  |
| --- | --- | --- |
| **INPUT**  User Input:  P = Passed, F = Failed  Age  eyesight exam(P/F)  written test(P/F)  Driving Test(P/F)  Medical fitness certificate available or not  For questions regarding passing/failure, only values acceptable are P and F. | **PROCESSING**  -Validate Age requirement (>= 18).  -Evaluate eyesight exam  -Evaluate written test  -Evaluate driving test with age(> 60 or <= 60)  -Evaluate Medical fitness certificate  Determine Eligiblity Status based on all the evaluation and validation criteria  -Ask if they want to recheck their eligiblity | **OUTPUT**  Display whether the user is eligible for driving test or not. |

**PSEUDOCODE**

START

DECLARE age

P = “Passed”

F = “Failed”

DECLARE eyesightExam

DECLARE writtenTest

DECLARE drivingTest

DECLARE medicalCertExists

DECLARE checkEligliblity

checkEligibility = “Yes”

REPEAT

PRINT “Enter age”

INPUT age

IF age >= 18 THEN

PRINT “Eyesight exam?(P/F)”

INPUT eyesightExam

IF eyesightExam == P THEN

PRINT “Written Test?(P/F)”

INPUT writtenTest

IF writtenTest == P THEN

PRINT “Driving test?(P/F)”

INPUT drivingTest

IF drivingTest == P THEN

IF age > 60 THEN

PRINT “Medical fitness certificate exists?”

INPUT medicalCertExists

IF medicalCertExists == yes THEN

PRINT “You are eligible”

ELSE

PRINT “You are not eligible”

ELSE

PRINT “You are eligible”

ELSE

PRINT “You are not eligible”

ELSE

PRINT “You need to retake the written test”

ELSE

PRINT “You might need a prescription for glasses”

ELSE

PRINT “ You are ineligible”

ENDIF

PRINT “Do you want to recheck your eligibility(Yes/No)?”

INPUT checkEligiblity

UNTIL

checkEligibility == “No”

END

**ALGORITHM**

1.Check the eligibility for driving license of the user.

2. Ask the user his age, if it is equal to or greater than 18 continue to ask the user about eyesight exam and written tests, if he passes each of them, continue to ask him about driving test.

3. If the user passes the driving test and is above 60, continue to ask him for his medical fitness certificate.

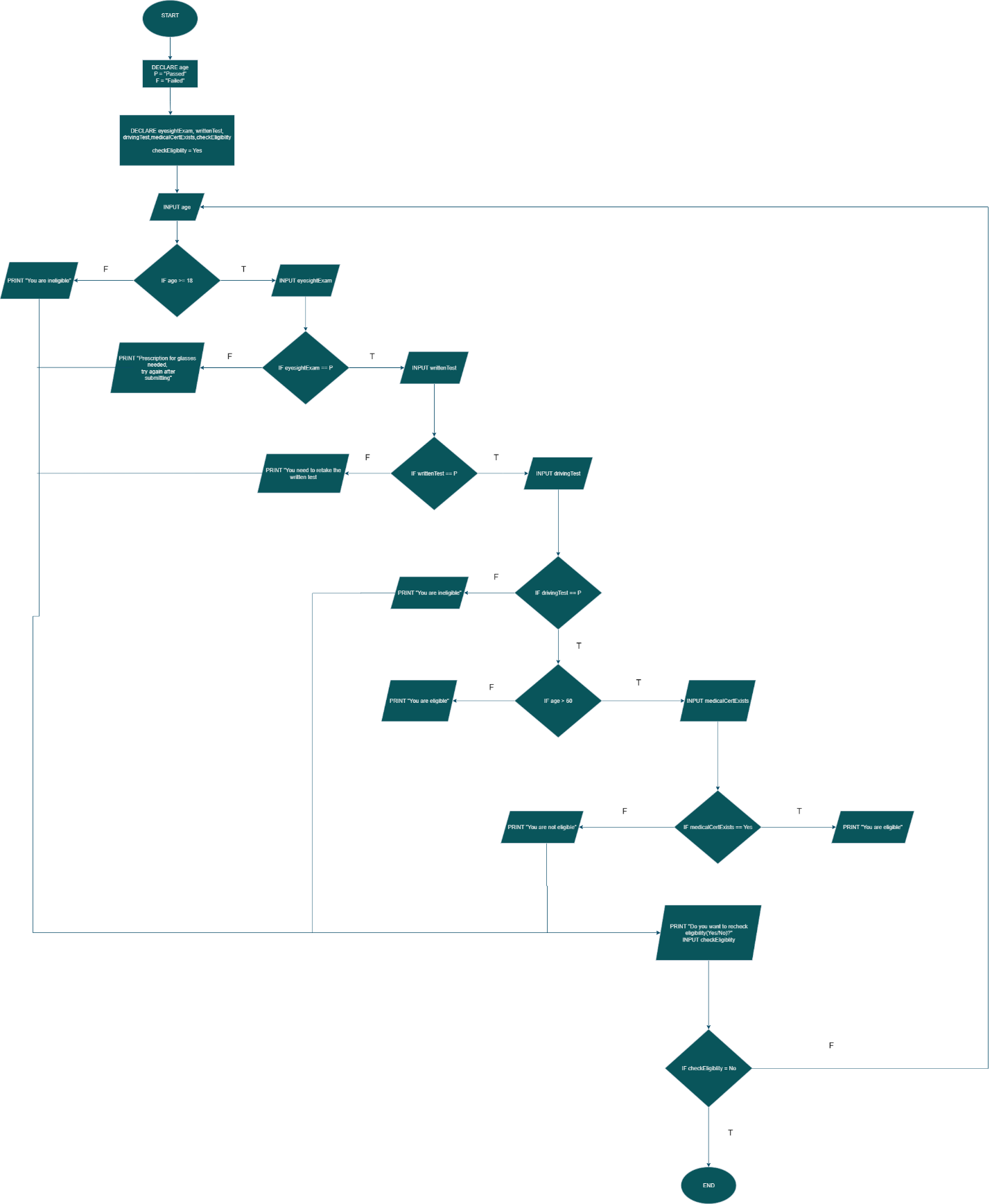
4. If medical fitness certificate exists, the user is eligible.

5.If age is below or equal to 60 and driving test has been passed, the user is eligible.

6. If driving test has been failed, the user is ineligible.

7.Ask the user to recheck if he wants.

**FLOWCHART**

****

Q4. PAC CHART

|  |  |
| --- | --- |
| **GIVEN INPUTS**  Poker hand is a sequence of 5 integers from 1 to 13  • 1 represents Ace,  • 2–10 represent cards numbered 2 through 10,  • 11 represents Jack,  • 12 represents Queen,  • 13 represents King.  A hand is a full house if it contains:  • three cards of one rank, and  • two cards of another rank.  User input of 5 numbers | **REQUIRED OUTPUT**  Display the poker hand(sequence of 5 integers) and whether it’s a full house or not |
| **PROCESSING REQUIRED**  Take user input of 5 numbers from 1 to 13  Check the frequency of each rank, when the user enters a number from a rank, its frequency increases by 1. If one rank has frequency of 3 and one rank has frequency of 2, it means that it’s a full house | **ALTERNATIVE SOLUTIONS**  Data structure and sorting(Advanced method) |

IPO CHART

|  |  |  |
| --- | --- | --- |
| **INPUT**  Integers that correspond to their respectives ranks and rank names.  User input of integer | **PROCESSING**  ENTER 5 integers from 1 to 13. COUNT the frequency of rank based on number entered.  IF there are three card of one rank and two cards of another, it is a full house | **OUTPUT**  DISPLAY the result to user  Telling whether it is a full house or not, with sequence of numbers. |

**Algorithm**

1.The user enters 5 numbers within the range 1 to 13.

2.Each rank has corresponding numbers or range of numbers with it.

3.When user enters a number, the frequency of rank that corresponds to that number increases.

4.After all 5 inputs, if one rank has a frequency of 3 and another has a frequency of 2, the user is informed that it is a full house poker hand, else tell the user that its not a full house poker hand.

**PSEUDOCODE.**

**START**

**ACECOUNT = 0**

**TWOCOUNT = 0**

**THREECOUNT = 0**

**FOURCOUNT = 0**

**FIVECOUNT = 0**

**SIXCOUNT = 0**

**SEVENCOUNT = 0**

**EIGHTCOUNT = 0**

**NINECOUNT = 0**

**TENCOUNT = 0**

**JACKCOUNT = 0**

**QUEENCOUNT = 0**

**KINGCOUNT = 0**

**PRINT "Enter 5 numbers, each ranging from 1 to 13"**

**INPUT num1**

**INPUT num2**

**INPUT num3**

**INPUT num4**

**INPUT num5**

**IF num1 = 1 THEN**

**ACECOUNT += 1**

**ELSE IF num1 = 2 THEN**

**TWOCOUNT += 1**

**ELSE IF num1 = 3 THEN**

**THREECOUNT += 1**

**ELSE IF num1 = 4 THEN**

**FOURCOUNT += 1**

**ELSE IF num1 = 5 THEN**

**FIVECOUNT += 1**

**ELSE IF num1 = 6 THEN**

**SIXCOUNT += 1**

**ELSE IF num1 = 7 THEN**

**SEVENCOUNT += 1**

**ELSE IF num1 = 8 THEN**

**EIGHTCOUNT += 1**

**ELSE IF num1 = 9 THEN**

**NINECOUNT += 1**

**ELSE IF num1 = 10 THEN**

**TENCOUNT += 1**

**ELSE IF num1 = 11 THEN**

**JACKCOUNT += 1**

**ELSE IF num1 = 12 THEN**

**QUEENCOUNT += 1**

**ELSE IF num1 = 13 THEN**

**KINGCOUNT += 1**

**IF num2 = 1 THEN**

**ACECOUNT += 1**

**ELSE IF num2 = 2 THEN**

**TWOCOUNT += 1**

**ELSE IF num2 = 3 THEN**

**THREECOUNT += 1**

**ELSE IF num2 = 4 THEN**

**FOURCOUNT += 1**

**ELSE IF num2 = 5 THEN**

**FIVECOUNT += 1**

**ELSE IF num2 = 6 THEN**

**SIXCOUNT += 1**

**ELSE IF num2 = 7 THEN**

**SEVENCOUNT += 1**

**ELSE IF num2 = 8 THEN**

**EIGHTCOUNT += 1**

**ELSE IF num2 = 9 THEN**

**NINECOUNT += 1**

**ELSE IF num2 = 10 THEN**

**TENCOUNT += 1**

**ELSE IF num2 = 11 THEN**

**JACKCOUNT += 1**

**ELSE IF num2 = 12 THEN**

**QUEENCOUNT += 1**

**ELSE IF num2 = 13 THEN**

**KINGCOUNT += 1**

**IF num3 = 1 THEN**

**ACECOUNT += 1**

**ELSE IF num3 = 2 THEN**

**TWOCOUNT += 1**

**ELSE IF num3 = 3 THEN**

**THREECOUNT += 1**

**ELSE IF num3 = 4 THEN**

**FOURCOUNT += 1**

**ELSE IF num3 = 5 THEN**

**FIVECOUNT += 1**

**ELSE IF num3 = 6 THEN**

**SIXCOUNT += 1**

**ELSE IF num3 = 7 THEN**

**SEVENCOUNT += 1**

**ELSE IF num3 = 8 THEN**

**EIGHTCOUNT += 1**

**ELSE IF num3 = 9 THEN**

**NINECOUNT += 1**

**ELSE IF num3 = 10 THEN**

**TENCOUNT += 1**

**ELSE IF num3 = 11 THEN**

**JACKCOUNT += 1**

**ELSE IF num3 = 12 THEN**

**QUEENCOUNT += 1**

**ELSE IF num3 = 13 THEN**

**KINGCOUNT += 1**

**IF num4 = 1 THEN**

**ACECOUNT += 1**

**ELSE IF num4 = 2 THEN**

**TWOCOUNT += 1**

**ELSE IF num4 = 3 THEN**

**THREECOUNT += 1**

**ELSE IF num4 = 4 THEN**

**FOURCOUNT += 1**

**ELSE IF num4 = 5 THEN**

**FIVECOUNT += 1**

**ELSE IF num4 = 6 THEN**

**SIXCOUNT += 1**

**ELSE IF num4 = 7 THEN**

**SEVENCOUNT += 1**

**ELSE IF num4 = 8 THEN**

**EIGHTCOUNT += 1**

**ELSE IF num4 = 9 THEN**

**NINECOUNT += 1**

**ELSE IF num4 = 10 THEN**

**TENCOUNT += 1**

**ELSE IF num4 = 11 THEN**

**JACKCOUNT += 1**

**ELSE IF num4 = 12 THEN**

**QUEENCOUNT += 1**

**ELSE IF num4 = 13 THEN**

**KINGCOUNT += 1**

**IF num5 = 1 THEN**

**ACECOUNT += 1**

**ELSE IF num5 = 2 THEN**

**TWOCOUNT += 1**

**ELSE IF num5 = 3 THEN**

**THREECOUNT += 1**

**ELSE IF num5 = 4 THEN**

**FOURCOUNT += 1**

**ELSE IF num5 = 5 THEN**

**FIVECOUNT += 1**

**ELSE IF num5 = 6 THEN**

**SIXCOUNT += 1**

**ELSE IF num5 = 7 THEN**

**SEVENCOUNT += 1**

**ELSE IF num5 = 8 THEN**

**EIGHTCOUNT += 1**

**ELSE IF num5 = 9 THEN**

**NINECOUNT += 1**

**ELSE IF num5 = 10 THEN**

**TENCOUNT += 1**

**ELSE IF num5 = 11 THEN**

**JACKCOUNT += 1**

**ELSE IF num5 = 12 THEN**

**QUEENCOUNT += 1**

**ELSE IF num5 = 13 THEN**

**KINGCOUNT += 1**

**THREEEXISTS = FALSE**

**TWOEXISTS = FALSE**

**IF ACECOUNT = 3 OR TWOCOUNT = 3 OR THREECOUNT = 3 OR FOURCOUNT = 3 OR FIVECOUNT = 3 OR SIXCOUNT = 3 OR SEVENCOUNT = 3 OR EIGHTCOUNT = 3 OR NINECOUNT = 3 OR TENCOUNT = 3 OR JACKCOUNT = 3 OR QUEENCOUNT = 3 OR KINGCOUNT = 3 THEN**

**THREEEXISTS = TRUE**

**ENDIF**

**IF ACECOUNT = 2 OR TWOCOUNT = 2 OR THREECOUNT = 2 OR FOURCOUNT = 2 OR FIVECOUNT = 2 OR SIXCOUNT = 2 OR SEVENCOUNT = 2 OR EIGHTCOUNT = 2 OR NINECOUNT = 2 OR TENCOUNT = 2 OR JACKCOUNT = 2 OR QUEENCOUNT = 2 OR KINGCOUNT = 2 THEN**

**TWOEXISTS = TRUE**

**ENDIF**

**IF TWOEXISTS = TRUE AND THREEEXISTS = TRUE THEN**

**PRINT (num1,num2,num3,num4,num5)**

**PRINT "Poker hand is full house"**

**ELSE**

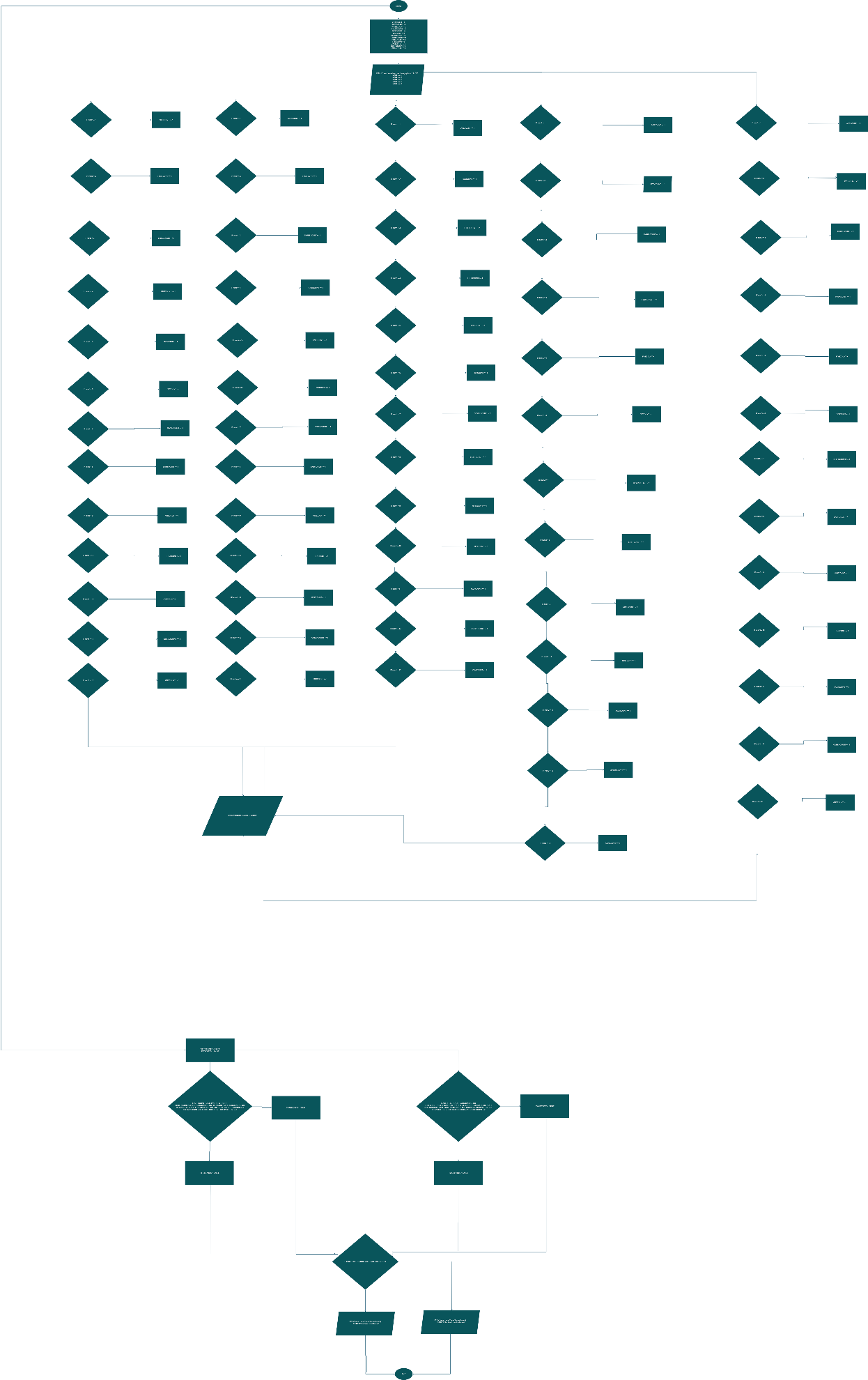
**PRINT (num1,num2,num3,num4,num5)**

**PRINT "Poker hand is not full house"**

**ENDIF**

**END**

**FLOWCHART**

****

Q5. PAC CHART

|  |  |
| --- | --- |
| **GIVEN INPUTS**  User input(1-digit integer only) | **REQUIRED OUTPUT**  Numbers and number of times user has input them |
| **PROCESSING REQUIRED**  Repeatedly ask user to enter a 1 digit integer until the user enters anything other a 1 digit integer from 0 to 9.  Count the number of times the integer was input | **ALTERNATIVE SOLUTIONS**  Ask user to enter the digit, then print its count, then again ask user to enter the digit again and print its count until the user enters anything other than 1 digit number.  In the end print the integer and its total number of occurrences. |

IPO CHART

|  |  |  |
| --- | --- | --- |
| **INPUT**  User Input of a 1 digit number from 0 to 9 | **PROCESSING**  ASK for user input  READ user input  IF Input is a single digit integer  INCREASE its count by 1  Ask for input again  IF Input is not a single digit integer, STOP asking for Input AND DISPLAY the integers and their respective counts. | **OUTPUT**  Integers from 0 to 9 and their respective counts(number of occurrences) |

**ALGORITHM**

1.Ask user to enter a 1 digit number.

2.Each time a user enters a number, increase that number’s count by 1.

3.Keep asking the user for a 1 digit number until the user enters a non 1 digit number.

4. Then, when the user enters a non 1 digit number, Display all the numbers and their number of occurrences or counts.

**PSEUDOCODE**

START

ZEROCOUNT = 0

ONECOUNT = 0

TWOCOUNT = 0

THREECOUNT = 0

FOURCOUNT = 0

FIVECOUNT = 0

SIXCOUNT = 0

SEVENCOUNT = 0

EIGHTCOUNT = 0

NINECOUNT = 0

DECLARE num

REPEAT

PRINT "Enter a 1 digit number"

INPUT num

IF IS\_NUMERIC(num) THEN

IF num = 0 THEN

ZEROCOUNT += 1

ELSE IF num = 1 THEN

ONECOUNT += 1

ELSE IF num = 2 THEN

TWOCOUNT += 1

ELSE IF num = 3 THEN

THREECOUNT += 1

ELSE IF num = 4 THEN

FOURCOUNT += 1

ELSE IF num = 5 THEN

FIVECOUNT += 1

ELSE IF num = 6 THEN

SIXCOUNT += 1

ELSE IF num = 7 THEN

SEVENCOUNT += 1

ELSE IF num = 8 THEN

EIGHTCOUNT += 1

ELSE IF num = 9 THEN

NINECOUNT += 1

ENDIF

ENDIF

UNTIL

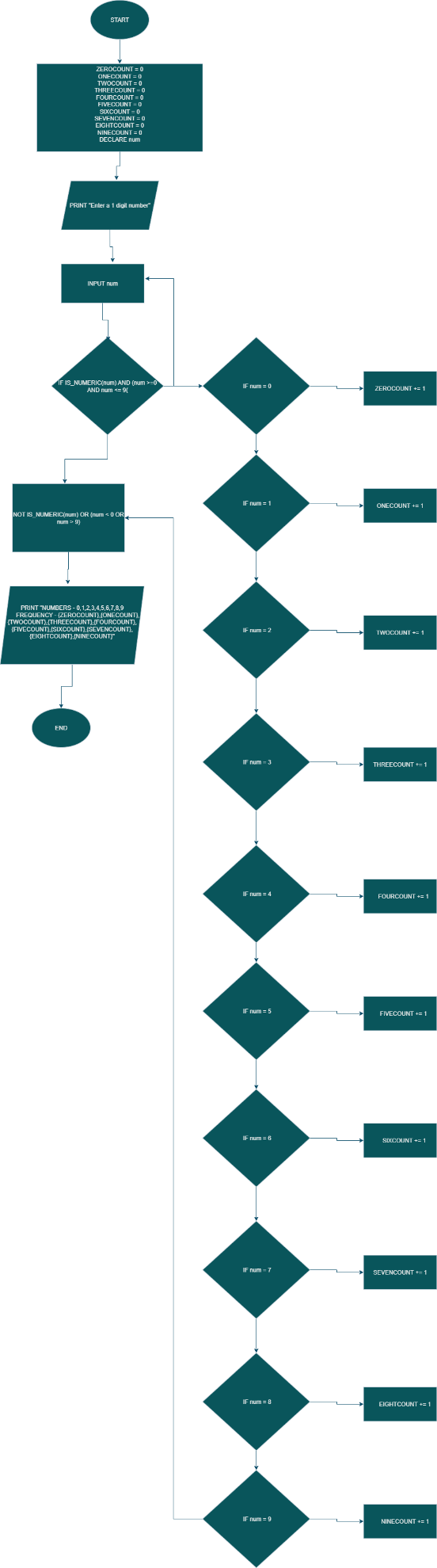
NOT IS\_NUMERIC(num) OR (num < 0 OR num > 9)

PRINT "NUMBERS - 0,1,2,3,4,5,6,7,8,9

FREQUENCY - {ZEROCOUNT},{ONECOUNT},{TWOCOUNT},{THREECOUNT},{FOURCOUNT},{FIVECOUNT},{SIXCOUNT},{SEVENCOUNT},{EIGHTCOUNT},{NINECOUNT}"

END

**FLOWCHART**

****