**WIX1002 Fundamentals of Programming**

**Tutorial 1 Problem Solving in Programming**

**Draw the Input Process Output (IPO) model and build the pseudocode, flow chart for**

**each of the problems:**

**Part I**

1. **Request two numbers from the user and print the multiplication of the numbers.**

**IPO Model:**

Input: Two numbers from the user

Process: Multiplication of the two numbers

Output: Result of multiplication

**Pseudocode:**

Start

DECLARE num1, num2, result AS REAL

DISPLAY "Enter the first number:”

INPUT num1 DISPLAY "Enter the second number: "

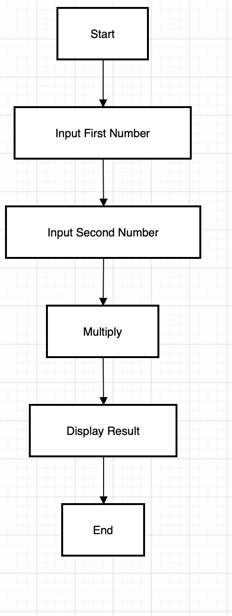
INPUT num2 result = num1 \* num2 DISPLAY "Multiplication of the given two numbers = "

Calculate the result

Display the result

End

**FLOWCHART:**

****

1. **Determine whether a random number is greater than 50.**

**IPO Model:**

|  |  |
| --- | --- |
| Input | Generate a random number between 1 and 100. |

|  |  |
| --- | --- |
| Process | Compare the random number with 50. |

|  |  |
| --- | --- |
| Output | Print whether the number is greater than, equal to, or less than 50. |
|  |  |
|  | **Pseudocode:** |

BEGIN

CREATE a Random object

GENERATE a random number between 1 and 100

IF number is greater than or equal to 50 THEN

PRINT "The number X is greater than or equal to 50"

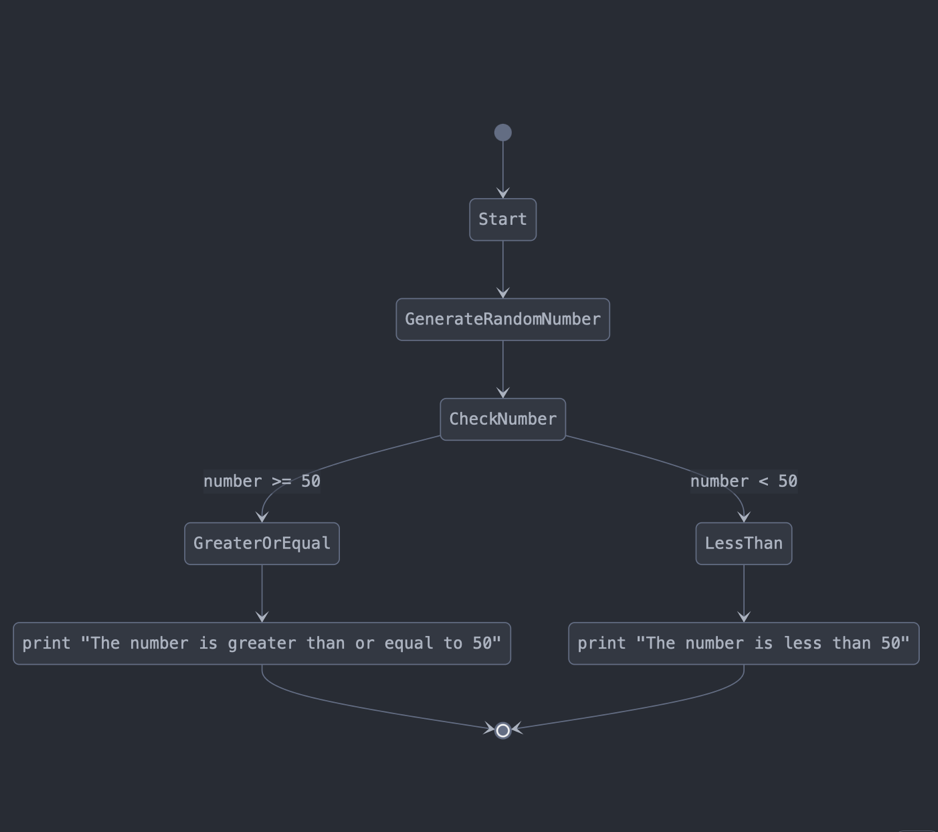
ELSE

PRINT "The number X is less than 50"

ENDIF

END

**FLOWCHART:**



**3. Print the pass/fail grade based on the mark entered by user. The passing mark is at**

**least 40.**

**IPO Model:**

|  |  |
| --- | --- |
| **Input** | User enters the obtained mark. |

|  |  |
| --- | --- |
| **Process** | Compare the mark with the passing mark (40). If the mark is 40 or higher, the student passes; otherwise, they fail. |

|  |  |
| --- | --- |
| **Output** | Display either "Congratulations! You passed" or "You failed in the examination." |
|  |  |

**Pseudocode:**

BEGIN

DISPLAY "Enter your obtained mark: "

READ mark

IF mark >= 40 THEN

DISPLAY "Congratulations! You are passed in the examination"

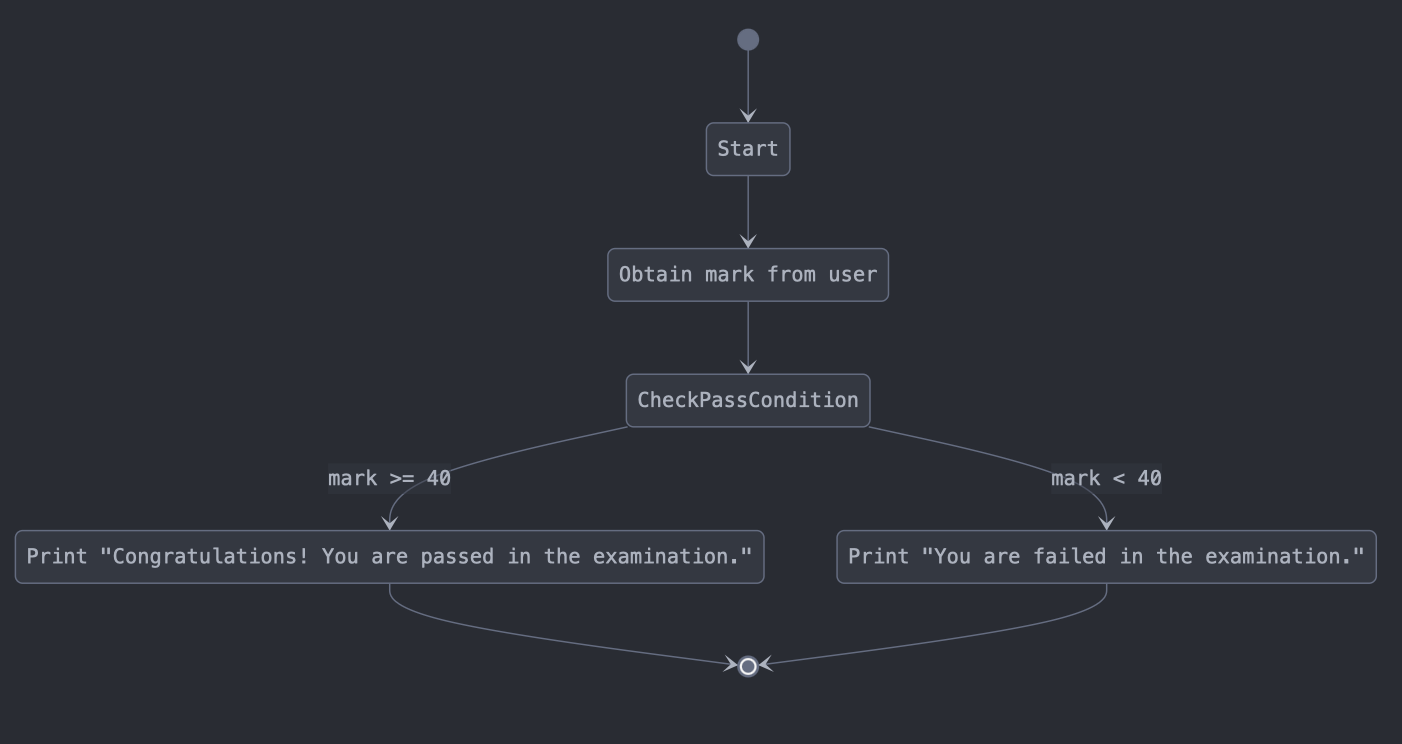
ELSE

DISPLAY "You are failed in the examination."

ENDIF

END

**FLOWCHART:**

****

**4. Print the results of the two players’ dice game.**

**IPO Model:**

|  |  |
| --- | --- |
| **Input** | User enters the obtained mark. |
| **Process** | Compare the mark with the passing mark (40). If the mark is 40 or higher, the student passes; otherwise, they fail. |
| **Output** | Display either "Congratulations! You passed" or "You failed in the examination." |
|  |  |

**Pseudocode:**

BEGIN

DISPLAY "Enter your obtained mark: "

READ mark

IF mark >= 40 THEN

DISPLAY "Congratulations! You are passed in the examination"

ELSE

DISPLAY "You are failed in the examination."

ENDIF

END

**FLOWCHART:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**5. Print the perimeter of a rectangle.**

**IPO Model:**

|  |  |
| --- | --- |
| **Input** | User enters the length and width of the rectangle. |

|  |  |
| --- | --- |
| **Process** | Calculate the perimeter using the formula: P=2×(length+width) |

|  |  |
| --- | --- |
| **Output** | Display the perimeter of the rectangle. |

**Pseudocode:**

BEGIN

DISPLAY "Enter the length of the rectangle: "

READ length

DISPLAY "Enter the width of the rectangle: "

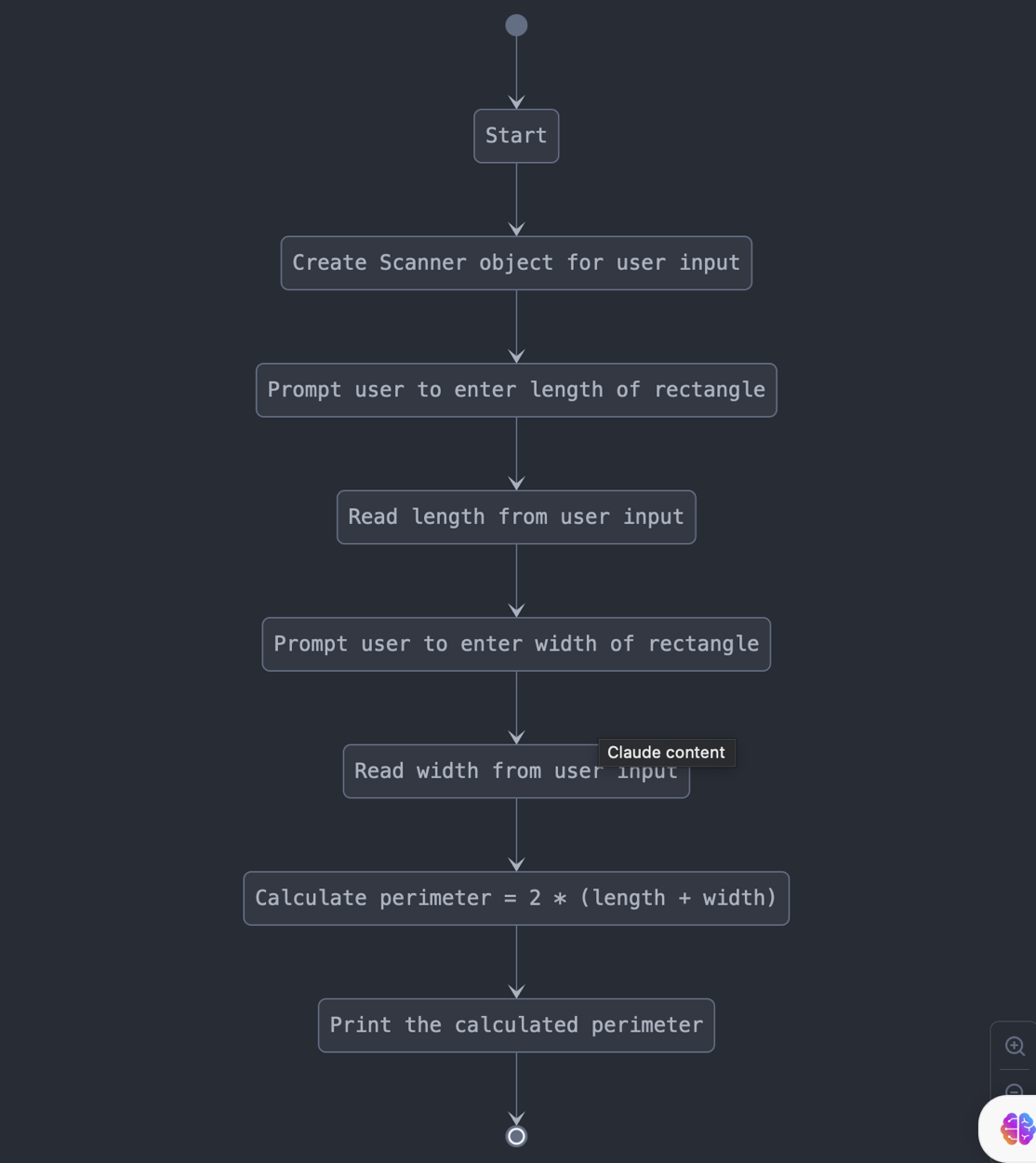
READ width

COMPUTE perimeter = 2 \* (length + width)

DISPLAY "The perimeter of the rectangle is: ", perimeter

END

**FLOWCHART:**

****

**6. Print the minimum number from 10 random numbers generated by computer.**

**IPO Model:**

|  |  |
| --- | --- |
| **Input** | Generate 10 random numbers between 0 and 999. |

|  |  |
| --- | --- |
| **Process** | Loop through the numbers, keep track of the smallest and largest values. |

|  |  |
| --- | --- |
| **Output** | Display the 10 generated numbers, the minimum number, and the maximum number. |
|  |  |

**Pseudocode:**

BEGIN

CREATE a Random number generator

DISPLAY "Generated numbers: "

SET min to a large number (1000)

SET max to a small number (0)

FOR i = 1 to 10 DO

GENERATE a random number between 0 and 999

DISPLAY the generated number

IF random number < min THEN

UPDATE min = random number

ENDIF

IF random number > max THEN

UPDATE max = random number

ENDIF

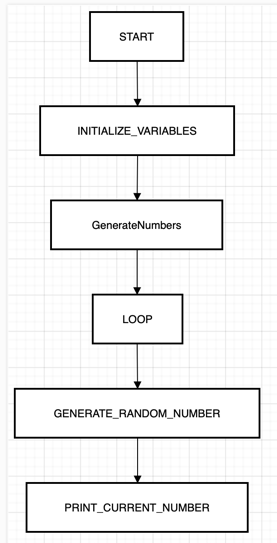
ENDFOR

DISPLAY "Minimum number: ", min

DISPLAY "Maximum number: ", max

END

**FLOWCHART:**



**7. Print the number of odd and even number from 10 random numbers generated by**

**computer. The random number must be from 10 – 100.**

**IPO Model:**

|  |  |
| --- | --- |
| **Input** | Generate 10 random numbers between 10 and 100. |

|  |  |
| --- | --- |
| **Process** | Loop through the numbers, check if each number is odd or even, and keep count. |

|  |  |
| --- | --- |
| **Output** | Display the 10 generated numbers, the count of even numbers, and the count of odd numbers. |

**Pseudocode:**

BEGIN

CREATE a Random number generator

INITIALIZE oddCount = 0, evenCount = 0

DISPLAY "Generated Numbers: "

FOR i = 1 to 10 DO

GENERATE a random number between 10 and 100

DISPLAY the generated number

IF random number MOD 2 == 0 THEN

INCREMENT evenCount

ELSE

INCREMENT oddCount

ENDIF

ENDFOR

DISPLAY "Total Number of even numbers: ", evenCount

DISPLAY "Total Number of odd numbers: ", oddCount

END

**FLOWCHART:**

**A diagram of a flowchart

AI-generated content may be incorrect.**

**Part II**

**8. Count the number of alphabet U and M from a sentence entered by user.**

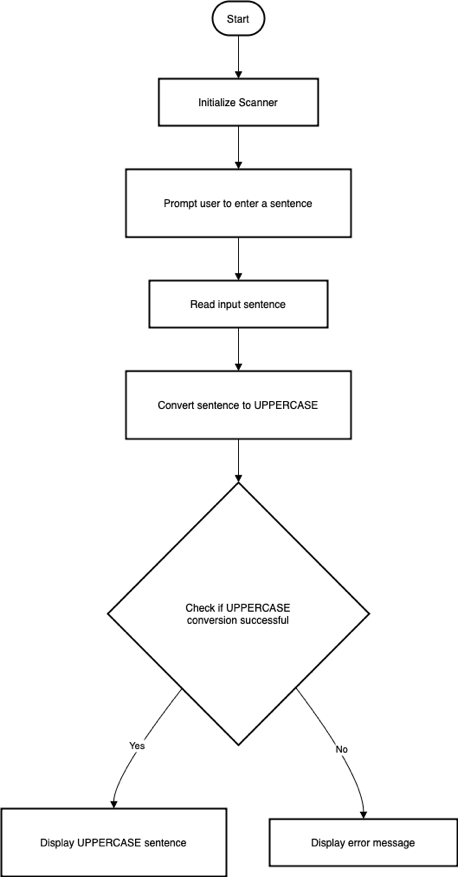
**IPO Model:**

|  |  |
| --- | --- |
| **Input** | User enters a sentence. |

|  |  |
| --- | --- |
| **Process** | Convert the sentence to uppercase. Loop through each character and count occurrences of 'U' and 'M'. |

|  |  |
| --- | --- |
| **Output** | Display the number of times 'U' and 'M' appear in the sentence. |
|  |  |
|  | **Pseudocode:**  BEGIN  DISPLAY "Please enter a sentence: "  READ sentence  CONVERT sentence to uppercase  INITIALIZE countU = 0, countM = 0  FOR each character in sentence DO  IF character == 'U' THEN  INCREMENT countU  ELSE IF character == 'M' THEN  INCREMENT countM  ENDIF  ENDFOR  DISPLAY "Number of 'U' in the sentence: ", countU  DISPLAY "Number of 'M' in the sentence: ", countM  END |

**FLOWCHART:**

****

**9. Display the frequency of a keyword from a web page.**

**IPO Model:**

|  |  |
| --- | --- |
| **Input** | Fetch the webpage content using Jsoup. Define a keyword to search. |

|  |  |
| --- | --- |
| **Process** | Extract visible text, convert it to lowercase, and count occurrences of the keyword. |

|  |  |
| --- | --- |
| **Output** | Display how many times the keyword appears on the webpage. |

**Pseudocode:**

BEGIN

TRY

SET url = "https://www.um.edu.my"

FETCH webpage content using Jsoup

EXTRACT visible text from the webpage and convert to lowercase

SET keyword = "um"

INITIALIZE count = 0

INITIALIZE index = 0

WHILE keyword is found in text starting from index

INCREMENT count

MOVE index forward by keyword length

ENDWHILE

DISPLAY "The keyword appears", count, "times."

CATCH IOException

DISPLAY "Error fetching or parsing the web page"

END

**FLOWCHART:**

A diagram of a network

AI-generated content may be incorrect.

**10. Display the number of female student from a random list of 100 students.**

**IPO Model:**

|  |  |
| --- | --- |
| **Input** | Generate random genders (0 = male, 1 = female) for 100 students. |

|  |  |
| --- | --- |
| **Process** | Loop through 100 students, assign a random gender, count the number of males and females. |

|  |  |
| --- | --- |
| **Output** | Display the number of female and male students. |

**Pseudocode:**

BEGIN

INITIALIZE totalStudents = 100

INITIALIZE femaleStudentsCount = 0

INITIALIZE maleStudentsCount = 0

FOR i = 1 to totalStudents DO

GENERATE a random number (0 or 1)

IF random number == 1 THEN

INCREMENT femaleStudentsCount

ELSE

INCREMENT maleStudentsCount

ENDIF

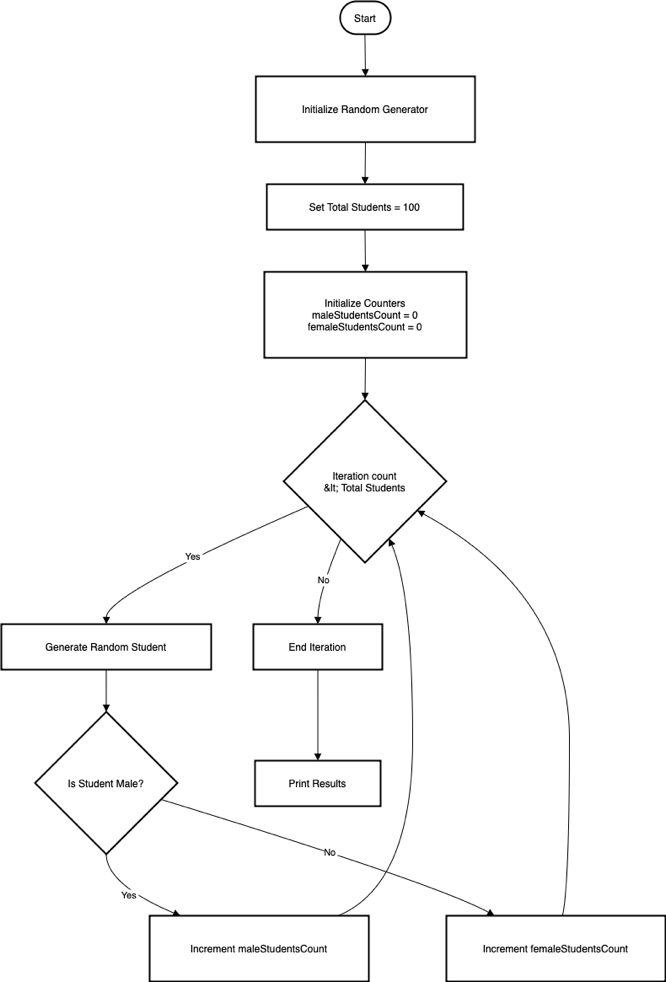
ENDFOR

DISPLAY "Number of female students: ", femaleStudentsCount

DISPLAY "Number of male students: ", maleStudentsCount

END

**FLOWCHART:**

****

**11. Display a list of 5 random numbers in descending order. (Sort)**

**IPO Model:**

|  |  |
| --- | --- |
| **Input** | Generate 5 random numbers (0 to 1000). |

|  |  |
| --- | --- |
| **Process** | Store the numbers in an array, sort them in descending order. |

|  |  |
| --- | --- |
| **Output** | Display the original numbers and the sorted numbers in descending order. |
|  |  |

**Pseudocode:**

BEGIN

DECLARE an array of size 5

FOR i = 0 to 4 DO

GENERATE a random number between 0 and 1000

STORE the number in the array

ENDFOR

DISPLAY "Original random numbers are: "

FOR each number in array DO

PRINT number

ENDFOR

SORT the array in descending order

DISPLAY "Random numbers in descending order: "

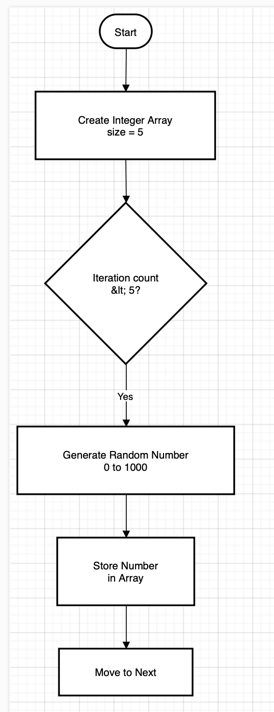
FOR each number in sorted array DO

PRINT number

ENDFOR

END

**FLOWCHART:**

****

**12. Guess a random number generated by computer.**

**IPO Model:**

|  |  |
| --- | --- |
| **Input** | User enters a guess for a randomly generated number (1–100). |

|  |  |
| --- | --- |
| **Process** | Compare the guess with the random number, provide hints, and count the attempts. |

|  |  |
| --- | --- |
| **Output** | Inform the user if the guess is too low, too high, or correct. Display the number of attempts when the correct guess is made. |

**Pseudocode:**

BEGIN

GENERATE a random number between 1 and 100

INITIALIZE attempts = 0

INITIALIZE guess = 0

DISPLAY "I have generated a random number between 1 and 100."

DISPLAY "Try to guess the number!"

WHILE guess is not equal to random number DO

PROMPT user to enter a guess

READ user input

INCREMENT attempts

IF guess is less than random number THEN

DISPLAY "Too low! Try again."

ELSE IF guess is greater than random number THEN

DISPLAY "Too high! Try again."

ELSE

DISPLAY "Congratulations! You guessed the number."

DISPLAY "It took you", attempts, "attempts."

ENDIF

ENDWHILE

END

**FLOWCHART:**

