**Project 1 Test Plan**

**Program Goals and Objectives**

The purpose of this program is to read an unspecified number of integers, determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros). The program has to end with the input 0.

**Program Functional Requirements**

1. The user needs to be prompted for integer inputs.
2. The input must be repeated infinitely until 0 is entered.
3. If the first entry is a 0, the program must output “No numbers are entered except 0”.
4. When 0 is finally entered, the program must output the number of positive and negative inputs, the total sum of the inputs and the average input value.

**Test Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Expected Result | Actual Result | Outcome (pass/fail) |
| 1 | 0 | No numbers are entered except 0 | No numbers are entered except 0 | pass |
| 2 | 2 3 4 5 0 | The number of positives is 4  The number of negatives is 0  The total is 14.0  The average is 3.5 | The number of positives is 4  The number of negatives is 0  The total is 14.0  The average is 3.5 | pass |
| 3 | -4 3 2 -1 0 | The number of positives is 2  The number of negatives is 2  The total is 0.0  The average is 0.0 | The number of positives is 2  The number of negatives is 2  The total is 0.0  The average is 0.0 | pass |
| 4 | -5 -1 word two 0 | Error message | Exception in thread "main" java.util.InputMismatchException | fail |

**Pseudocode / Flowchart**

Function Main

Declare Double counter

Declare Int counterNeg

Declare Int counterPos

Declare Double average

Declare Double num

While true

Output "Enter an integer, the input ends if it is 0: "

Input num

Assign counter = counter + num

If num = 0 && counterNeg + counterPos = 0

Ouput "No numbers are entered except 0"

break

Else if num = 0

Assign average = counter / (counterNeg + counterPos)

Output "The number of positives is " + counterPos

Output "The number of negatives is " + counterNeg

Output "The total is " + counter

Output "The average is " + average

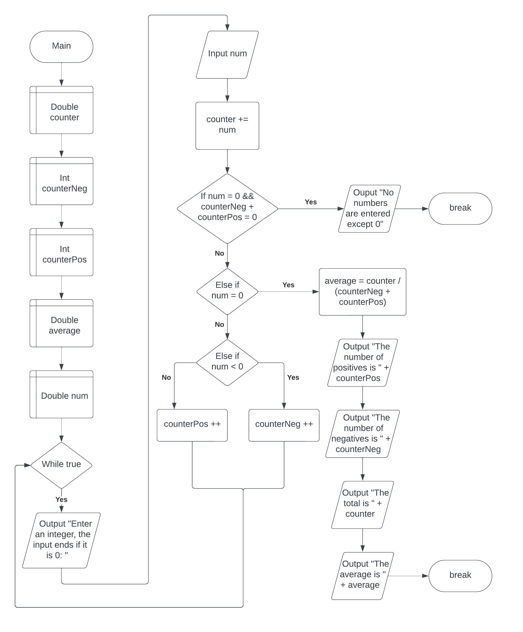
break

Else if num < 0

Assign counterNeg = counterNeg + 1

Else

counterPos = counterPos + 1



**Project 2 Test Plan**

**Program Goals and Objectives**

The purpose of this program is to read exactly 12 digits of inputs used to identify books, calculate a 13th digit using a given formula and output the full 13 digits called an ISBN-13.

**Program Functional Requirements**

1. The user needs to be prompted to enter a 12-element long string consisting of digits.
2. If the string is not 12 elements or is not all digits, the program must output the string is an invalid input.
3. The program must calculate the 13th element using the formula d13 = 10 - (d1 + 3d2 + d3 + 3d4 + d5 + 3d6 + d7 + 3d8 + d9 + 3d10 + d11 + 3d12) % 10.
4. If d13 is 10, the program must replace it with 0.
5. The program must combine the first 12 digits with the calculated 13th digit into a string
6. The program must output “The ISBN-13 number is “ && full 13 digit ISBN-13

**Test Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Expected Result | Actual Result | Outcome (pass/fail) |
| 1 | 978013213080 | The ISBN-13 number is 9780132130806 | The ISBN-13 number is 9780132130806 | pass |
| 2 | 978013213079 | The ISBN-13 number is 9780132130790 | The ISBN-13 number is 9780132130790 | pass |
| 3 | 9780132130 | 9780132130 is an invalid input | 9780132130 is an invalid input | pass |
| 4 | nine78013213080 | nine78013213080 is an invalid input | nine78013213080 is an invalid input | pass |
| 5 | 978013f13080 | 978013f13080 is an invalid input | 978013f13080 is an invalid input | pass |

**Pseudocode / Flowchart**

Function Main

Declare String isbn

Declare Boolean invalid

Declare IntArray d[]

Assign d[] length 12

Output "Enter the first 12 digits of an ISBN-13 as a string: "

Input isbn

If isbn.length == 12

For int i = 0; i < 12; i++

If not Character.isDigit(isbn.charAt(i))

Ouput isbn + " is an invalid input"

Assign invalid = true

break

Assign d[i] = Integer.parseInt(isbn.substring(i, i + 1))

Else

Output isbn + " is an invalid input"

Assign invalid = true

If not invalid

Declare Int isbn13

Assign isbn13 = 10 - (d[0] + 3\*d[1] + d[2] + 3\*d[3] + d[4] + 3\*d[5] + d[6] + 3\*d[7] + d[8] + 3\*d[9] +

d[10] + 3\*d[11]) % 10

If isbn13 = 10

isbn13 = 0

Declare String fullIsbn

Assign fullIsbn = isbn + isbn13

Output "The ISBN-13 number is " + fullIsbn

