Assignment 2

Enrique Saracho Felix

100406980

CPSC 1150

10/06/2023

# Exercise 1

## Program CountingOnes

**Purpose:** Count the number of 1s in a binary number which is converted from a decimal number entered by the user.

**Input:** Decimal number of type integer. The program outputs a message to the user to enter an integer number between 0 and 99. If the value entered by the user doesn’t fit the criteria, then it will display an error message, and prompt the user to enter the value again. The value will be stored in a variable called *num*.

**Output:** Decimal number of type integer. The program will display the *count* variable containing the number of 1s in the binary equivalent of the input value.

**Calculations:** Conversion of the user input from decimal to binary. We divide in a loop the input by 2 until the answer is 0. We also keep track of the remainder of each division.

For the remainder: num % 2 == 1

For the division: num = (int)(num / 2)

**Variables:** num: Type integer. Contains the input value.

count: Type integer. Counts the number of 1s in the binary number.

**Constants:** Not needed.

**Flowchart:**

A picture containing text, diagram, screenshot, font

Description automatically generatedA picture containing text, diagram, line, font

Description automatically generated

**Trace tables:**

Algorithm trace table:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **num** | **count** | **Output** |
| 0 | -5 | 0 | Error |
| 1 | 14 | 4 | 4 |
| 2 | 100 | 0 | Error |
| 3 | 0 | 0 | 0 |
| 4 | 5 | 2 | 2 |
| 5 | 30 | 4 | 4 |

Loop trace table (example with num = 5):

|  |  |  |
| --- | --- | --- |
| **Iteration** | **count** | **num** |
| 0 | 0 | **5** |
| 1 | 1 | 2 |
| 2 | 1 | 1 |
| 3 | **2** | 0 |

Loop trace table (example with num = 30):

|  |  |  |
| --- | --- | --- |
| **Iteration** | **count** | **num** |
| 0 | 0 | **30** |
| 1 | 0 | 15 |
| 2 | 1 | 7 |
| 3 | 2 | 3 |
| 4 | 3 | 1 |
| 5 | **4** | 0 |

Loop trace table (example with num = 14):

|  |  |  |
| --- | --- | --- |
| **Iteration** | **count** | **num** |
| 0 | 0 | **14** |
| 1 | 0 | 7 |
| 2 | 1 | 3 |
| 3 | 2 | 1 |
| 4 | **3** | 0 |

# Exercise 2

## Program Palindrome

**File name:** (file’s relative path)

**Purpose:** To generate a random three-digit integer and determine if it’s a palindrome number.

**Packages:** (list of imported packages)

**Limitations:** (input it can’t handle, list of possible error messages, round-off error)

**Bugs:** (list of unfixed bugs)

**Input:** Not needed.

**Output:** A string containing the random generated number and a Boolean value representing whether it’s a palindrome or not.

**Pseudocode:**

Algorithm (program name)

START

1. Declare variables int *num*.
2. Generate a random integer between 100 and 999 and assign it to *num*.
3. If num % 10 = rounded down(num / 100)
   1. Print num is a palindrome.
4. Else
   1. Print num is not a palindrome.

END (program name)

**Trace table:**

|  |  |  |
| --- | --- | --- |
| **num** | **num % 10 = num / 100** | **Output** |
| 123 | False | 123 is not a palindrome. |
| 454 | True | 454 is a palindrome. |