Aatif Sayed

Professor Andy Guna

CS111: Introduction to Computer Science

20 September 2019

**Pseudocode Assignment**

1. “*Gas*” Problem
   1. Inputs:
      1. Price per gallon
      2. Number of gallons of gas that were purchased
      3. Whether person pays by cash or credit
   2. Outputs:
      1. The price a person will pay for gas
   3. Error Conditions:
      1. Zero (0) being inputted
      2. If no payment method is specified (neither cash nor credit)
   4. Pseudocode Algorithm:

READ pricePerGallon

READ numGallons

READ cash(boolean)

READ credit(boolean)

IF pricePerGallon <= 0 OR numGallons <= 0 THEN

DISPLAY errorMessage: "Neither the price per gallon nor the number of gallons purchased can be 0 or negative."

ENDIF

IF cash(boolean) is false AND credit(boolean) is false THEN

DISPLAY errorMessage: “Please specify a payment method.”

ENDIF

COMPUTE totalPrice AS pricePerGallon \* numGallons

IF credit(boolean) is true THEN

COMPUTE totalPrice AS totalPrice \* 1.1

ENDIF

DISPLAY totalPrice

* 1. Test cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **pricePerGallon** | **numGallons** | **cash(boolean)** | **credit(boolean)** | **totalPrice** |
| 0 | - | - | - | ERROR |
| - | -2 | - | - | ERROR |
| 3 | 5 | true | false | 15 |
| 2 | 8 | false | true | 17.6 |
| - | - | false | false | ERROR |
| -2.5 | - | - | - | ERROR |
| - | -4 | - | - | ERROR |

* 1. Minimum number of operations: 7
  2. Maximum number of operations: 12

1. “*Train Ticket for One Person*” Problem
   1. Inputs:
      1. Person’s age
      2. Whether the ticket is bought at the station (yes/no)
      3. Whether the ticket is bought on the train (yes/no)
   2. Outputs:
      1. Price of a ticket
   3. Error Conditions:
      1. Age being less than 0 (no negative numbers allowed as inputs)
      2. Age being above 120
      3. If no location of where the ticket was bought is specified (neither at station nor in train)
   4. Pseudocode Algorithm:

READ age

READ station(boolean)

READ train(boolean)

IF station(boolean) is false AND train(boolean) is false THEN

DISPLAY errorMessage: “Please specify a location where the ticket was bought (either at the station or on the train).”

IF age < 0 OR age > 120 THEN

DISPLAY errorMessage: "Age cannot be below 0 or above 120."

ELSE IF age < 7 THEN

SET price TO 0

ELSE IF age > 65 THEN

SET price TO 7.5

ELSE

SET price TO 13.2

ENDIF

IF train(boolean) is true THEN

COMPUTE price AS price \* 1.2

ENDIF

DISPLAY price

* 1. Test cases:

|  |  |  |  |
| --- | --- | --- | --- |
| **age** | **station(boolean)** | **train(boolean)** | **price** |
| -2 | - | - | ERROR |
| 145 | - | - | ERROR |
| - | false | false | ERROR |
| 4 | true | false | 0 |
| 82 | false | true | 9 |
| 43 | false | true | 15.84 |
| 43 | true | false | 13.2 |

* 1. Minimum number of operations: 6
  2. Maximum number of operations: 13

1. “*WCS111 FM*”Problem
   1. Inputs:
      1. Number of hours spent programming per month
   2. Outputs:
      1. The prize that is received
   3. Error Conditions:
      1. Number of hours inputted is less than 0 (negative)
   4. Pseudocode Algorithm:

READ hours

IF hours < 0 THEN

DISPLAY errorMessage: "Number of hours spent programming cannot be a negative number."

ELSE IF hours = 0 THEN

Set prize TO nothing

ELSE IF hours >= 1 AND hours <= 5 THEN

SET prize TO tshirt

ELSE IF hours >= 6 AND hours <= 400 THEN

IF (hours + 1) % 10 = 0 THEN

SET prize TO laptop

ENDIF

IF hours % 2 = 0 THEN

SET prize TO hat

ENDIF

IF hours % 3 = 0 THEN

SET prize TO tv

ENDIF

IF hours % 2 = 0 AND hours % 3 = 0 THEN

SET prize TO hat and tv

ENDIF

IF (hours + 1) % 10 = 0 AND hours % 3 = 0 THEN

SET prize TO laptop and tv

ENDIF

IF (hours + 1) % 10 != 0 AND hours % 2 != 0 AND hours % 3 != 0 THEN

SET prize TO nothing

ENDIF

ELSE

SET prize TO cat

ENDIF

DISPLAY prize

* 1. Test cases:

|  |  |
| --- | --- |
| **hours** | **prize** |
| -2 | ERROR |
| 0 | nothing |
| 4 | shirt |
| 19 | laptop |
| 8 | hat |
| 15 | tv |
| 9 | laptop and tv |
| 6 | hat and tv |
| 7 | nothing |
| 561 | cat |

* 1. Minimum number of operations: 3
  2. Maximum number of operations: 26

1. “*Lucky Sevens*” Problem
   1. Inputs:
      1. Whole number
   2. Outputs:
      1. The number of sevens that appear in the given whole number
   3. Error Conditions:
      1. Whole number inputted cannot be negative
   4. Pseudocode Algorithm:

READ number

SET count AS 0

IF number < 0 THEN

DISPLAY errorMessage: "Number inputted must be a positive integer."

ENDIF

WHILE number != 0

IF number % 10 = 7 THEN

ADD 1 TO count

ENDIF

COMPUTE number AS number / 10

ENDWHILE

DISPLAY count

* 1. Test cases:

|  |  |
| --- | --- |
| **number** | **count** |
| -7 | ERROR |
| 37227773 | 4 |
| 70707 | 3 |
| 7 | 1 |
| 1023 | 0 |

* 1. Minimum number of operations: 4
  2. Maximum number of operations: 5 + 3n (where n is number of digits in the given whole number)