Unit 3

Practice IV

Algorithms

**Goal of the practice**

Know how to use invocation of algorithms and what is the purpose of them. Use predefined functions and create new functions to solve problems. Notion of reusability.

Understand the difference between parameters and arguments.

Understand the definition of the scopes of variables.

1. What are the advantages and disadvantages of modular programming. Explain in words how to model with modules a problem where you have to read from the keyboard a series of numbers (which can be stored in a list) and display the 3 major numbers and 3 minor ones.

Advantages: divide a complex problem in small simpler problems. It small solutions could be uses for solve other problems.

Disadvantajes: It may be required to write more code, and it could be slower at the runtime

Read a serie of numbers

Find the N major(s)

Find the N minor(s)

1. Count the number of words, separated by one or more spaces, from a telegram ending in point. It can be assumed that the user enters character by character the telegram or the complete sequence, which is more comfortable to propose a solution.

algorithm main

var String : text = ""

var Number : count = 0

text = readText('.')

count = countWords('.', text)

PRINT("Number of words is " + count)

end algorithm

algorithm readText (Char : endText) : String

var String : text = ""

var Char : character = ''

WHILE (character != endText) DO

READ(character)

text = text + character

END WHILE

return text

end algorithm

algorithm countWords (Char : endText, String text) : Number

var Number : count = 0

var Char : character = nextChar(text)

var Char : previousCharacter = ' '

WHILE (character != endText) DO

IF(previousCharacter == ' ' AND character != ' ') THEN

count = count + 1

END IF

previousCharacter = character

character = nextChar(text)

END WHILE

return count

end algorithm

1. Given the following algorithms:

algorithm a1

var Number : x

x 🡨 10

print (x \* 5)

x 🡨 a2(x\*2)

x 🡨 x \* 2

a2(x)

end algorithm

algorithm a2(Number : x) : Number

print (x)

x 🡨 x / 2

print (x)

return x

end algorithm

Identify the scopes of each variable. Track the execution of a1 indicating what is displayed by console. Idem for the execution (independently) of the algorithm a2.

|  |  |
| --- | --- |
| a1 | a2 |
| 50 |  |
|  | 20 |
|  | 10 |
|  | 20 |
|  | 10 |

1. Develop an algorithm for a dice game. The player must bet on a number between 1 and 6 (keyboard reading), then you must simulate the roll of a dice and finally inform the player if he has won or lost (print on screen).

algorithm main

var Number : value = 0

READ(value)

IF(DiceGame(value)) THEN

PRINT("You won.")

ELSE

PRINT("You lose.")

END IF

end algorithm

algorithm DiceGame (Number : value) : Boolean

var Boolean : result = false

var Number : dice = RANDOM(1,6)

if(dice == value)

result = true

return result

end algorithm

1. Write an algorithm to invert a string of characters.

algorithm InvertString (String : text) : String

var String : result = text

var Number : lenght = LENGHT(text)

var Number : counter = 0

WHILE (counter < (lenght/2)) DO

REPLACE(result, counter, CharIn(text, (lenght-1)-counter))

REPLACE(result, (counter-1)-i, CharIn(text, counter))

counter = counter + 1

END WHILE

return result

end algorithm

1. Explain what the following algorithm performs

algorithm toInt (String : cad)

var Number : long,i = 0,pot = 1,res = 0

var String : c

long = length (cad)

while (i < long) do

c = charIn(cad,long - i - 1)

res = res + ord(c) \* pot

pot = pot \* 10

i = i + 1

end while

return res

en algorithm

Where **ord** is a function that receives a character and returns its value in numeric format. This way it is compatible to work with arithmetic operations.

Converts a string value in a number

1. Propose a modular solution to the problem of determining if a number is prime.

Ask if the number is 1 2 or 3

If the number if five or more, find if it is prime

1. Write a sub-algorithm that has a parameter of type number that represents a year and determines whether the year is leap year or not.

algorithm isYearLeap (Number : year) : Boolean

var Boolean : result = false

IF(year MOD 4 == 0) THEN

result = true

ENDIF

return result

end algorithm

1. Write a sub-algorithm to determine the number of days of a month of a year, the latter two parameters of the sub-algorithm.

algorithm daysOfAMonth (Number : year, Number : month) : Number

var Number : result = 0

SWITCH (month) :

CASE 1:

result = 31

CASE 2:

IF (isYearLeap(year)) THEN

result = 29

ELSE

result = 28

ENDIF

CASE 3:

result = 31

CASE 4:

result = 30

CASE 5:

result = 31

CASE 6:

result = 30

CASE 7:

result = 31

CASE 8:

result = 31

CASE 9:

result = 30

CASE 10:

result = 31

CASE 11:

result = 30

CASE 12:

result = 31

DEFAULT:

result = 0

END SWITCH

return result

end algorithm

1. A meteorological station collects rain data for a month and year of a determined locality. An algorithm must be written asking the user to enter a month and a year, and allow for each day of that month in that year, enter the millimeters of water that were recorded (in case of not having registered a rain one day is entered 0) . The amount of total millimeters of water precipitated in that month must be shown, the maximum precipitation recorded for a day and on what day it was given, and finally if it rained two days in a row.

Ask for day and year

Get the number of days for that daysOfAMonth

Ask for the rain data for each day

Find the sum

Find the maximum and its position

Find if rained > 0 two times in a row