Trevor Fune

CSC 242

Professor Guha

28 Aug 2024

**Assignment 5 Pseudocode**

Spell Check Pseudocode

This pseudocode describes how to check the spelling of all words in a file utilizing a dictionary file and outputs all words not in the dictionary.

Declare input-stream variable named dictionary

Open the dictionary file in dictionary

If file doesn’t open:

Output “Unable to open dictionary file”

Define a vector of strings called words

Define a string called word

For each word in the dictionary file:

Append the word to the words vector

Close the dictionary file

Declare input-stream variable named words\_to\_check

Open the file to be checked (“WordsToCheck.txt”) in words\_to\_check

If file doesn’t open:

Output “Unable to open file to check spelling.”

For each word in words\_to\_check:

If the word is not contained in the words vector

Print the word

Country Populations Pseudocode

This pseudocode describes how to read country population data from a file and use it to calculate total population.

Create void function named process\_line:

Purpose: processes line containing country name and population into separate variables

Parameter named line: the string being processed

Parameter named country: name of country in line

Parameter named population: population of country

Create istringstream named strm that contains line

Create char variable named ch

For every character in strm

If character is not white space

Append character to country

If character is a white space

Skip to the next non-white space character

If character is not a digit

Append a space to country

Append character to country

Else

Unget the character

Place the whole number into population

Break out of function

Main Function:

Declare input-stream variable named input\_file

Open the world population file (“worldpop.txt”) in input\_file

If the file doesn’t open:

Output “Unable to open world population file.”

Create a string variable named line

Create a floating-point variable named total\_population

For every line in input\_file:

Create a string variable named country

Create an integer variable named population

Make function call to process\_line (line, country, population)

If country is not the European Union

Add population to total\_population

Close the file

Output the total population excluding the European Union

Keyword Cipher Pseudocode

This pseudocode describes how to encrypt or decrypt a file using a keyword cipher.

Create constant string variable named ALPHABET with all uppercase letters

Create string function named create\_encrypted\_alphabet

Purpose: Create encrypted alphabet using keyword and remaining letters in reverse order

Parameter named keyword: keyword used for cipher

Return: a string containing the encrypted alphabet

Create string named encrypted\_alphabet

Create Boolean array of size 26 named used to hold used letters

For every character in keyword

Convert character to uppercase

If character is a letter and is not in the used array

Append character to encrypted\_alphabet

Update the element representing the current character in used to true

Create string named remaining\_letters

For every character in alphabet

If the character hasn’t been used yet

Append the character to remaining\_letters

Reverse remaining\_letters

Append remaining\_letters to encrypted\_alphabet

Return encrypted\_alphabet

Create void function named encrypt\_file

Purpose: Encrypt a file using an encrypted alphabet

Parameter named input: ifstream of input file

Parameter named output: ofstream of output file

Parameter named encrypted\_alphabet: string containing encrypted alphabet

Create string named line

For every line in input

Create string named encrypted\_line

For every character in line

If character is a letter

Convert character to uppercase

Find position of character in ALPHABET

Append encrypted\_alphabet[pos] to encrypted\_line

Else

Append character to encrypted\_line

Send encrypted\_line to output

Create void function named decrypt\_file

Purpose: Decrypt a file using an encrypted alphabet

Parameter named input: ifstream of input file

Parameter named output: ofstream of output file

Parameter named encrypted\_alphabet: string containing encrypted alphabet

Create string named line

For every line in input

Create string named decrypted\_line

For every character in line

If character is a letter

Convert character to uppercase

Find position of character in encrypted\_alphabet

Append ALPHABET[pos] to decrypted\_line

Else

Append character to decrypted\_line

Send decrypted\_line to output

Main Function:

If number of command line arguments does not equal 5

Output: “Usage: keywordcipher.cpp <-e|-d> -k<keyword> <input\_file> <output\_file>

Exit out of program

Create string named mode and set equal to argv[1]

Create Boolean variable named encryption\_mode and set equal to true

If mode is “-e”

Keep encryption\_mode equal to true

Else if mode is “-d”

Set encryption\_mode to false

Else

Output “Invalid mode: use ‘-e’ or ‘d’”

Exit out of program

Create string named keyword\_arg and set equal to argv[2]

Create string named keyword and set equal to substring of argv[2] after “-k”

If keyword\_arg doesn’t have “-k” or a keyword

Output “Invalid keyword: use -k<keyword>

Exit out of program

Create ifstream named in\_file

Open argv[3] in in\_file

If in\_file fails to open

Output “Unable to open input file”

Exit out of program

Create ofstream named out\_file

Open argv[4] in out\_file

If out\_file fails to open

Output “Unable to open output file”

Exit out of program

Create string named encrypted\_alphabet using the function create\_encrypted\_alphabet

If encryption\_mode is true

Use the function encrypt\_file

If encryption\_mode is false

Use the function decrypt\_file