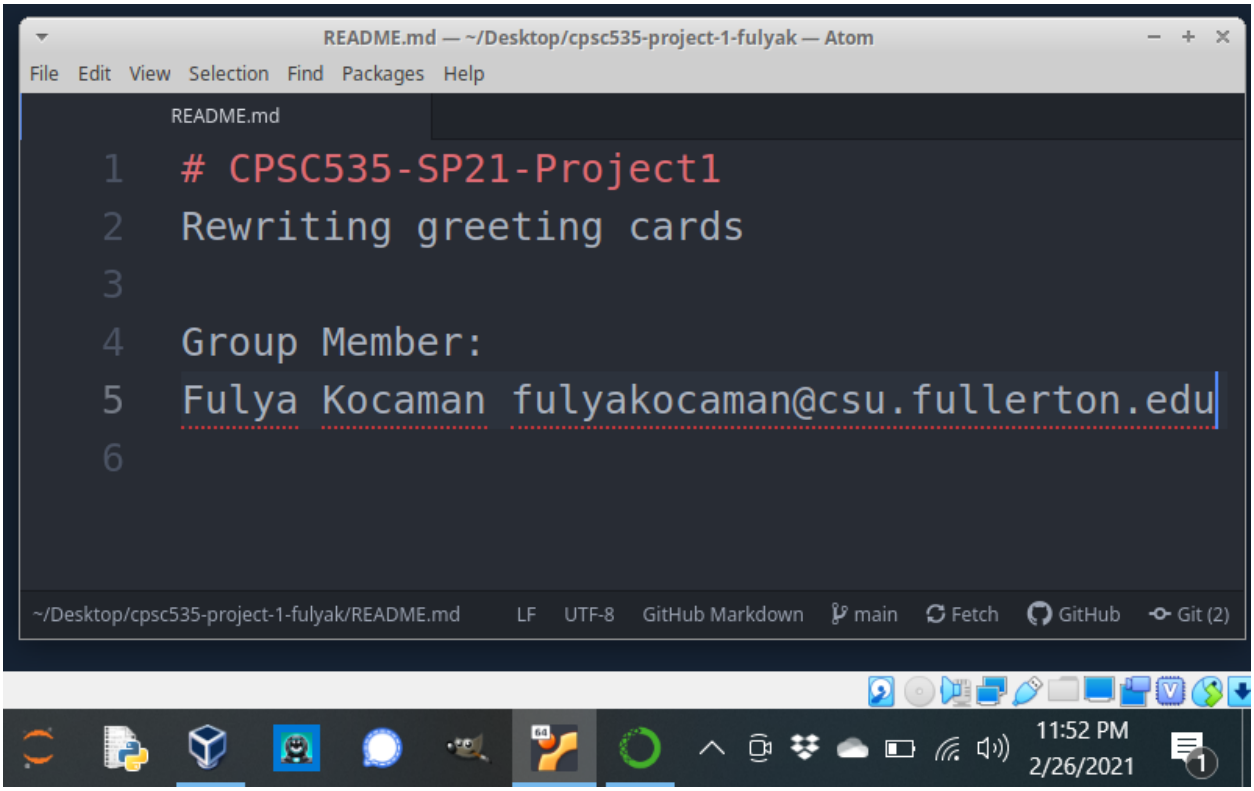
In this project I will be designing and implementing Rewriting Greeting Ecards algorithm related to strings using C++. This algorithm uses a string S of length N>0, and a list of M pairs of strings LS, each pair representing a string to be replaced and the second being the actual replacement, displays the new string R that is obtained by replacing every occurrence of the first string in each pair with the corresponding second string in the pair. The string matching must be case sensitive.



1. **Pseudocode:**

Input: string S of length N>0 and list of M pairs of strings LS

Output: the new string R that is obtained by replacing every occurrence of the first string in each pair with the corresponding second string in the pair.

```
def numWords(S):
  num = 0
  string wordRead
  open file S
   while !file.eof():
     file >> wordRead
     ++num
```

```
    endwhile
  close file
 return num
enddef


def vector<string> getVec (S, wordSize):
  // Creates array of vectors to store the strings from the file
  vector<string> vec
  // Stores each word read from a file
  string wordRead
  open fileS S
    for i = 0 to wordSize – 1 do:
      fileS >> wordRead
      vec.push_back(wordRead)
     endfor
  close fileS
 return vec
enddef

def map<string, string> getStringPairs (LS):
  // Stores each sentence read from a file
  string sentenceRead
  // Counts the number of lines read from a file
  numLineRead = 0
  index = 1
  /*(key, value) pairs. The key is a string to be replaced the value is the
  actual string replacement */
  map<string, string> pairs
  string key // a string to be replaced
  string value // the actual string replacement
  open fileLS LS
    while (getline(fileLS, sentenceRead)):
      key = sentenceRead
      if (numLineRead % 2 == 0):
        getline(fileLS, sentenceRead)
        value = sentenceRead
        index++
      endif
      pairs[key] = value
    endwhile
  close fileLS
 return pairs
enddef
```

```
def main(int argc, char **argv):
  if (argc < 3): // Sanity check -- make sure the user provided all of the required arguments
    fprintf(stderr,
          "USAGE: %s <StringS FILE NAME> <StringPairsLS FILE NAME> \n", argv[0])
    exit(1)
   endif
  // Stores the file name of the input string S
  string stringSFileName = argv[1]
  // Stores the file name of the input pairs of string LS
  string stringPairsLSFileName = argv[2]
  // The size of the vector of words
  int size = numWords(stringSFileName)
  // Vector of strings read from the stringS file
  vector<string> myVec = getVec(stringSFileName, size)
  // Pairs of strings read from the stringPairsLS file
  map<string, string> stringPairsLS = getStringPairs(stringPairsLSFileName)
  for std::vector<string>::iterator it = myVec.begin(); it != myVec.end(); ++it:
    string word = *it
    for std::map<string, string>::iterator it2 = stringPairsLS.begin(); it2 != stringPairsLS.end(); ++it2:
      int found = word.find(it2->first)
      if (found != string::npos):
        word.replace(word.find(it2->first), it2->first.length(), it2->second)
        *it = word
        break // If replaced, no need to search more
      endif
    endfor
  endfor
  // Resulting string
  string stringR
  for std::vector<string>::iterator it = myVec.begin(); it != myVec.end(); ++it:
    stringR = stringR + *it  + " "
  endfor
  stringR += '\0' // Adding the null terminator
  fprintf(stderr, "%s\n", stringR.c_str())
endmain
```

2. How to Run the Code:

C++ language is used in this project. From the Linux terminal:

To compile the greetingsCards.cpp use the command: g++ greetingsCards.cpp -o greetingsCards

To run the program, use the command:
./greetingsCards <StringS FILE NAME> <StringPairsLS FILE NAME>

, where <StringS FILE NAME> is the name of the file containing the string to be replaced and <StringPairsLS FILE NAME> is the name of the file containing the pairs of strings, each pair representing a string to be replaced and the second being the actual replacement.

3. Snapshots of the Code Executing for the Three Given Examples

Example 1:

Input:

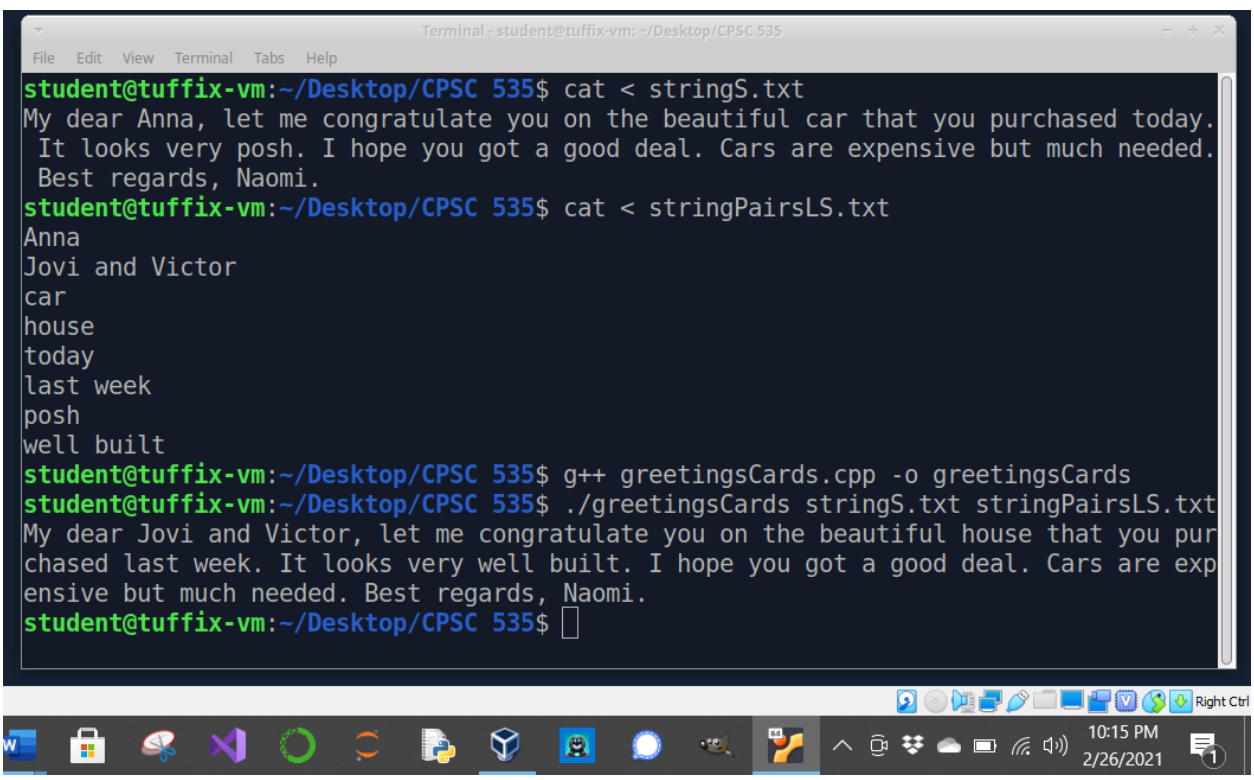
S[] = “My dear Anna, let me congratulate you on the beautiful car that you purchased today. It looks very posh. I hope you got a good deal. Cars are expensive but much needed. Best regards, Naomi.”

N = 147

M = 4

LS[] = { {“Anna”, “Jovi and Victor”} , { “car”, “house”}, {“today”, “last week”}, {“posh”, “well built”}}

The Example 1 Output:



Example 2:

Input:

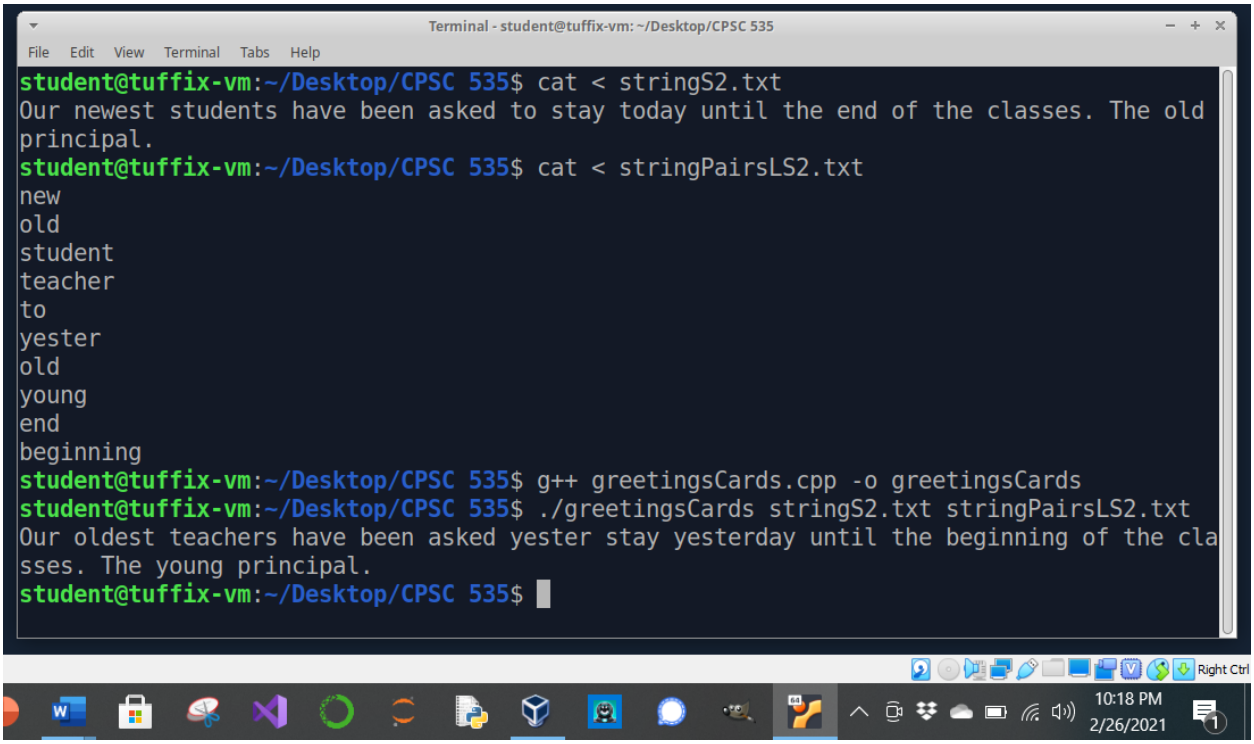
S[] = “Our newest students have been asked to stay today until the end of the classes. The old principal.”

N = 95

M = 5

LS[] = { {“new”, “old”} , { “student”, “teacher”}, {“to”, “yester”}, {“old”, “young”}, {“end”, “beginning”}}

The Example 2 Output:



Example 3:

Input:

S[] = “Our newest students have been asked to stay today until the end of the classes. The old principal.”

N = 95

M = 5

LS[] = {{"old", "young"}, {"student", "teacher"}, {"to", "yester"}, {"end", "beginning"}, {"new", "old"}}

The Example 3 Output:

Please see the next page of the example 3 output