Ashley Chang

504449890

Discussion 1K

Project 5: I’ve Got a Secret

a. I ran into a variety of very difficult obstacles while creating this code, namely working with more than one crib word. After finding a match for one crib word, I found it difficult to align the next crib words with the corresponding words in the cipher text. I was eventually able to solve this problem, though my resulting code was likely overly complicated. Also, it was difficult to move from line to line if a crib was not found, though I solved the problem with an additional while loop. Furthermore, I was unable to figure out how to check that the cipher words and crib words matched one-to-one and did not use different translations for the same character. At some point, I also ran into some problems in which cout and cerr lines would alter my program so that inserting and removing them would make my program go from running incorrectly to correctly; I assumed this was a result of accessing some undefined variables, but was unable to understand where I had gone wrong.

b.

Create a 2D new array to store the cipher text: separateMessages

Count how many messages (indicated by \n or ‘\0’) in cipher text

Store messages in corresponding rows of new 2D array

Screen every character of separateMessages

If detect a non alphabetic character, then replace with a space

(Clean 2D array so only words and (multiple) spaces)

Create a 3D array to store individual words in messages of ciphertext: splitMessageWord ([msgNumber][word][letter])

For every row of separateMessages indicating a message line,

Store letters of separateMessages in corresponding row of   
 splitMessageWord

If encounter a space followed by alphabetic character, create new row to

indicate new word

Create a 2D array to store cleaned crib: cleanupCrib (each word separated by a single space)

For every letter in crib until zero byte,

If the indicated character is alphabetic,

Put in position of new array

Else if the indicated character is non-alphabetic but the next character is,

Add a space to the new array

Otherwise, if indicated character is non-alphabetic & next character is nonalphabetic,

Increment position in crib until find alphabetic character or position of the last non-alphabetic character

When hit zero byte, stop forming cleanupCrib.

Screen cleanupCrib for spaces or zero bytes

Count how many words are in crib

Create a final 2D array to store crib so each word is a unique row: recordCrib

For every letter in cleanupCrib,

Store the character into a row of recordCrib

If find end of word (indicated by space), move to next row

recordCrib should have rows equal to number of words in crib

Match crib to ciphertext: Find the first message in splitMessageWord that has enough words to fit words in crib

While the indicated message is able to fit the crib,

Mark the first word in the message: msgCursor

While there is enough space from msgCursor to the end of the message,

Check if word at msgCursor is the same length as 1st crib word

Mark possible match

Check if following words have the same length as all of  
 following crib words and possible match

If following words are not matching in length, not a possible   
 match. Return to original msgCursor and continue search

If all of crib words have a length match, begin creating key

Create an array with 26 spaces

Starting from position where length match was first found,

Convert letter in message to integer

If spot in array corresponding to integer is   
 empty,

Make sure crib character was not already used

Otherwise, break out of loop

If valid, store the letter from message into array

If spot in array corresponding to integer is not  
 empty and the letter from the message does   
 not correspond with what is already there,   
 Start over and keep searching for crib   
 match in msg starting from original

msgCursor

If word is at msgCursor is not the same length as 1st crib word

Move on to next word (msgCursor++)

If msgCursor reaches the end of a message, move to the next message and continue search

Go through ciphertext and decode according to key:

Create a new array to print out translation: decipherment

For every message of the cipher

Go through every letter

If the cipher letter is a non alphabetic character,

Store the cipher letter in the decipherment array

Else if the cipher letter is an alphabetic character,

If the cipher letter corresponds to a position in key that has a zero byte inside,

Store the cipher letter in the decipherment array

If the cipher letter corresponds to a position in key that already has a crib character inside,

Store the key letter in the decipherment array

For every message of the cipher text,

Print the corresponding decipherment array with the finished translations.

c. List of Test Data

|  |  |
| --- | --- |
| Type of Test Data | Test Data |
| One crib word | (“Hirdd ejsy zu drvtry od.\nO'z fodvtrry.\n”, “secret”)   * Crib is found in cipher text   (“Hirdd ejsy zu drvtry od.\nO'z fodvtrry.\n”, “booooo”)   * Crib has length match but is not one-to-one, so no match   (“Hirdd ejsy zu drvtry od.\nO'z fodvtrry.\n”, “man”)   * No length match |
| Multiple crib words | (“Hirdd ejsy zu drvtry od.\nO'z fodvtrry.\n”, “my secret”)   * Crib is found in cipher text   (“urt olwd”, “dog paws”)   * Number of crib words match number of words in cipher text |
| No new lines | (“xpaab opmv, “hello dear”)   * Number of crib words match number of words in cipher text * Crib found in cipher text   (“O ziofa zit Zktqlxkt gy Doeiossofrq ol wxkotr lgdtvitkt of dn wqeanqkr.” , “treasure of michillinda”)   * Multiple crib words for one cipher text message * Crib found in cipher text   (“O ziofa zit Zktqlxkt gy Doeiossofrq ol wxkotr lgdtvitkt of dn wqeanqkr.” , “hello”)   * Crib not found in cipher text (but is length match) |
| Multiple new lines | (“Tftdn lxwdqkoft lhgzztr of Ltezgk zvtfzn-ltctf. \n  O ziofa zit Zktqlxkt gy Doeiossofrq ol wxkotr lgdtvitkt of dn wqeanqkr.\n  Ror ngx itqk ziqz Sgktzzq ol ugofu gxz vozi Dqkexl?\n  el 31 ol eiqsstfuofu!\n” , “treasure of michillinda”)   * Multiple messages in cipher text with multiple crib words * Special characters included in cipher text * Crib found in cipher text   (“Tftdn lxwdqkoft lhgzztr of Ltezgk zvtfzn-ltctf. \n  O ziofa zit Zktqlxkt gy Doeiossofrq ol wxkotr lgdtvitkt of dn wqeanqkr.\n  Ror ngx itqk ziqz Sgktzzq ol ugofu gxz vozi Dqkexl?\n  el 31 ol eiqsstfuofu!\n” , “michillinda”)   * Multiple messages in cipher text with one crib word * Special characters included in cipher text |
| Crib includes non-alphabetic, special letters | (“Hirdd ejsy zu drvtry od.\nO'z fodvtrry.\n”, “bo8903”)   * Single crib word with possible matches   (“Hirdd ejsy zu drvtry od.\nO'z fodvtrry.\n”, “boo90//clouds”)   * Multiple crib words with special characters with no full length matches |
| Crib is longer than cipher text | (“Hirdd ejsy zu ”, “Guess what my secret is”)   * Single cipher message * First three words in crib correspond with unique key characters   (“Hirdd ejsy zu drvtry od.\nO'z fodvtrry.\n el 31 ol eiqsstfuofu!\”, “why does the chicken jump over the fence and fly in the sky”)   * Multiple cipher messages * No length match with crib |
| Crib has no alphabetical characters or has no words | (“Hirdd ejsy zu drvtry od.\nO'z fodvtrry.\n”, “23098414”)   * Crib is digits only * One cipher message   (“Tftdn lxwdqkoft lhgzztr of Ltezgk zvtfzn-ltctf. \n  O ziofa zit Zktqlxkt gy Doeiossofrq ol wxkotr lgdtvitkt of dn wqeanqkr.\n  Ror ngx itqk ziqz Sgktzzq ol ugofu gxz vozi Dqkexl?\n  el 31 ol eiqsstfuofu!\n” , “”)   * Crib is empty * Multiple cipher messages |
| Cipher text is empty or has only special characters | (“”,””)   * Cipher text and crib is empty   (“”, “wow”)   * Single crib word   (“”, “egg pancake”)   * Multiple crib words   (“2039848&&64736”,”moo”)   * Single crib word   (“2039848&&64736”, “such fun”)   * Multiple crib words |

\*Highlighted test cases do not work