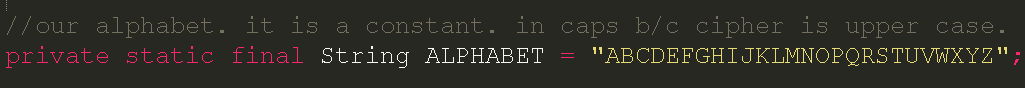
Kevin Siraki

Professor Boctor

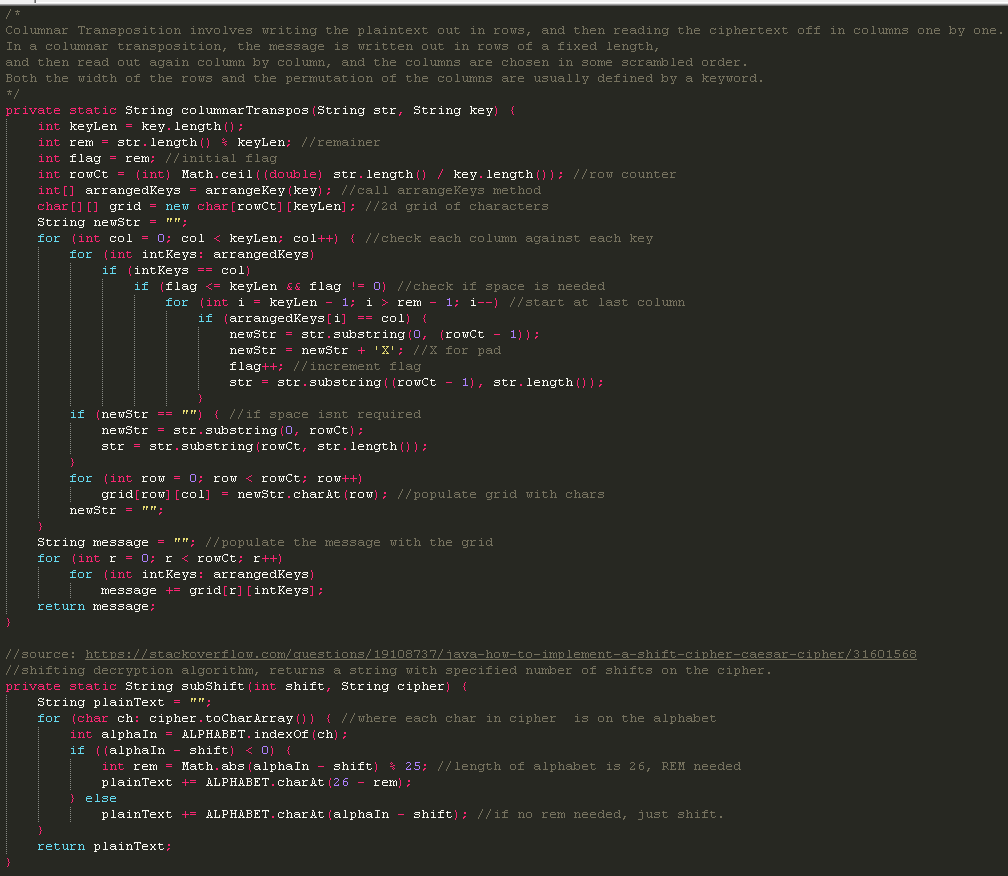
COMP 424

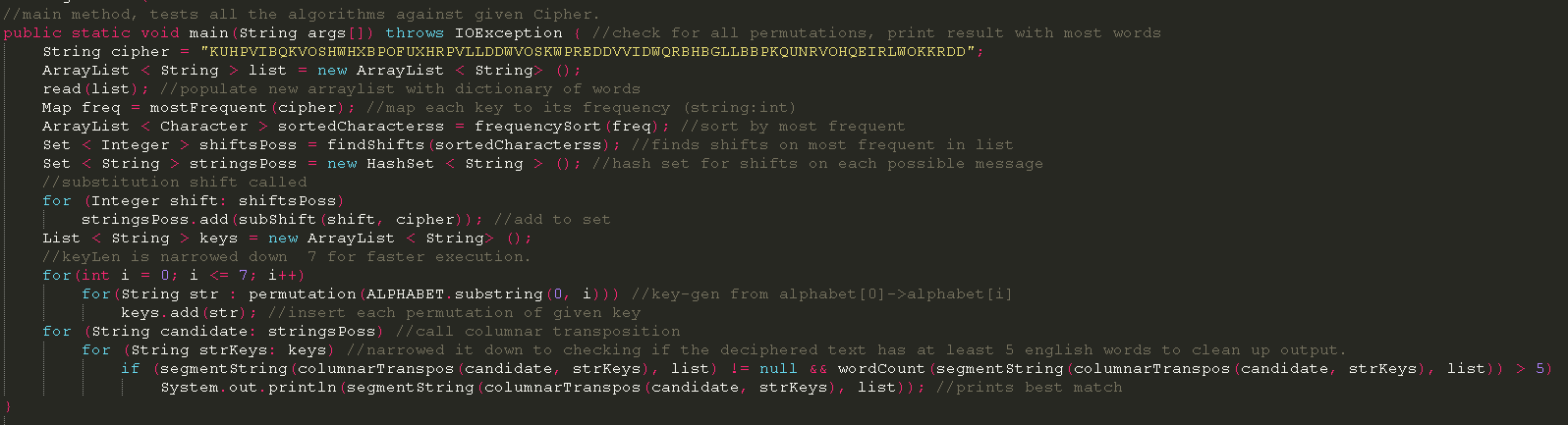
1 Oct. 2021

Cipher-text Decryption Techniques

 For this project, the task was to take a secret message of the length 77 and decipher it in order to retrieve the original message. We were originally presented with the given information that we must utilize columnar transposition and shift substitution in order to retrieve the message. Moreover, I was able to deduce that the key length was below the nominal value of 10 we were presented with because I was able to decrypt the message with keys of length 7 and the original length was 77. This proved to drastically reduce the runtime of the program and allowed for faster decryption. Moreover, knowing that the original message cannot contain any symbols or numbers was helpful in designing the algorithms themselves. To tackle this assignment, I chose Java as my language because I feel it has the necessary built-in types to easily solve this problem without sacrificing runtime with a fully interpreted yet simpler language like Python. Since we know the original message is in English and is in capital letters, our alphabet can be initialized as follows.

To begin with, the algorithm of Columnar Transposition basically consisted of a nested for loop that checks each column of the grid against each key that is passed in. Next, we started at the last column of the grid and went through the entire grid and populated it with the proper characters and return it at the end. In laymen’s terms, Columnar Transport requires us to write the plaintext into rows and read off the cipher text column by column. The rows will be of fixed length and will be read out again column by column, but the columns are in a randomized order. There is a predefined value that sets the width of the rows and columns themselves.

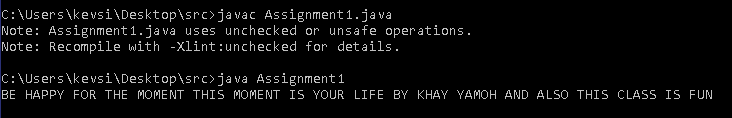
 Furthermore, the substitution shift (AKA Caesar’s Shift) algorithm is an algorithm that returns a string with a specified number of shifts on the cipher, both of which are passed in as arguments to the method. Essentially, Caesar’s Shift on the string “Hi” with a shift value of 1 would return “Ij”, as we shifted the string by 1 letter. This algorithm combined with putting the text in a table and rearranging the columns (columnar transposition), allows us to decrypt the given input string and retrieve the message. We will discuss some other helper methods that were required to get both of these key algorithms running in the following paragraph.

To elaborate, we needed a few helper methods to get Columnar Transposition and Caesars shift to work. Firstly, I wrote a method called “findShifts” which essentially lets us know how many shifts occur in a list. Next, the method “segmentString” will basically just split a string with spaces between words that are found in the Dictionary.txt file. This file just contains 1000 of the most common words in the English language. In essence, this algorithm works recursively by storing part of the word as the prefix, and stores the part after the space in a suffix variable that is then passed into the next recursive call. The next method I implemented was called “mostFrequent”. This method just maps a frequency value to each character of the cipher string and returns said map at the end. The method “frequencySort” takes this map and sorts it by the frequency key from least to most common. Next up is the merge method, which simply merges a character in a possible position in a string, then appends it to the given array list. The permutation method uses this function to generate a permutation of a given “ArrayList”. It takes the last character of the word and places it at the end of the permutation of the rest of the word. Lastly, the method “arrangeKey” just arranges an “ArrayList” based on keys, and returns an integer array of their positions. It uses the built-in sort method in Java and iterates over the array in the outer loop and iterates over the string in the inner loop. If the index values match the character positions in the string, we append the position to the array. Finally, the simpler helper methods such as “wordExits”, “wordCount”, and read just check if a word exists in an “ArrayList”, count the number of words in an “ArrayList” by seeing how many spaces occur, and read a Dictionary.txt file into an “ArrayList” respectively. We call all of these methods in the main method and use the “wordCount” method to only print the string with the most common words found in order to avoid printing excessive amounts of decrypted ciphers. As aforementioned, I limited the key length to 7 as that seemed to yield the proper result and anything higher would just take longer to execute. This made sense since the original string was 77 characters in length. All of the methods above were used to find candidates that are closest to an English sentence, and the output was limited to the single best one with wordCount.

To conclude, with the given string: “KUHPVIBQKVOSHWHXBPOFUXHRPVLLDDWVOSKWPREDDVVIDWQRBHBGLLBBPKQUNRVOHQEIRLWOKKRDD”,

Our final decrypted message was:

“BE HAPPY FOR THE MOMENT THIS MOMENT IS YOUR LIFE BY KHAY YAMOH AND ALSO THIS CLASS IS FUN”.

The proof of this can be seen from the following screenshot:

We can see that all of the aforementioned algorithms worked properly and we were able to achieve the goal of decrypting the message.

Works Cited

262588213843476. “1,000 Most Common US English Words.” *Gist*, https://gist.github.com/deekayen/4148741.

“Caesar Cipher in Cryptography.” *GeeksforGeeks*, 6 Apr. 2021, https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/.

“How to Merge Two Arraylists in Java.” *HowToDoInJava*, 30 Aug. 2020, https://howtodoinjava.com/java/collections/arraylist/merge-arraylists/.

IT\_PhilicIT\_Philic 20122 gold badges55 silver badges1515 bronze badges, et al. “How to Find the Most Frequently Occurring Character in a String with Java?” *Stack Overflow*, 1 May 1962, https://stackoverflow.com/questions/21750365/how-to-find-the-most-frequently-occurring-character-in-a-string-with-java.

Mhi. “Columnar Transposition Cipher with Key.” *Code 4 Life*, 1 Jan. 1970, https://programmingcode4life.blogspot.com/2015/09/columnar-transposition-cipher.html.

Mhi. “Columnar Transposition Cipher with Key.” *Code 4 Life*, 1 Jan. 1970, https://programmingcode4life.blogspot.com/2015/09/columnar-transposition-cipher.html.

*Print All Permutation of a String Using Arraylist in Java*, https://www.tutorialspoint.com/print-all-permutation-of-a-string-using-arraylist-in-java.

says:, Satvik Todankar, and Sandeep Patidar says: “Java String: String Functions in Java with Examples.” *Edureka*, 30 Sept. 2021, https://www.edureka.co/blog/java-string/.

Sophia AliSophia Ali 19122 gold badges55 silver badges1414 bronze badges, et al. “Java, How to Implement a Shift Cipher (Caesar Cipher).” *Stack Overflow*, 1 Jan. 1962, https://stackoverflow.com/questions/19108737/java-how-to-implement-a-shift-cipher-caesar-cipher/31601568.

“Sort Elements by Frequency: Set 5 (Using Java Map).” *GeeksforGeeks*, 11 Dec. 2018, https://www.geeksforgeeks.org/sort-elements-by-frequency-set-5-using-java-map/.