**Questions within the problem:**

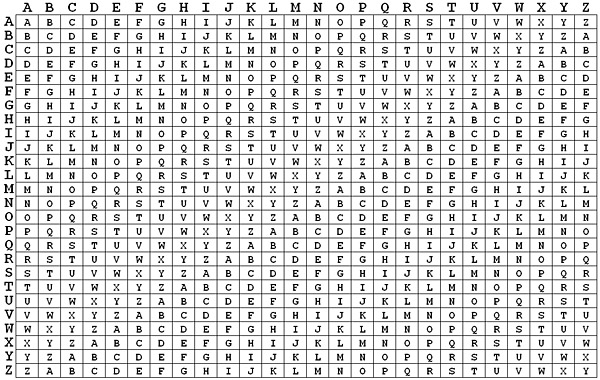
The main problems that were being attempted to be solved through this problem were the ability to utilize ascii conversions from a string input. The other problem is that of having the user enter a keyword that would be repeated with the same length as that of the user entered text.

**Solution to the problem:**

The approach to the problem was that of understanding basics of having the user enter a string that would be used to have ascii code referenced for encryption. From this understanding, it was simple to look back at unit 12 for references on how encryption and decryption was handled. And from understanding that, I decided to utilize for loop that would run throughout the provided length of the text given by the user. And while it goes through, it would calculate the ascii value for the char conversion. Once there, it would convert the value and insert back into a string, where it would be encrypted. The same approach would be followed with the decryption, but in the opposite way, where it would subtract from the ascii value.

**Calculations & Algorithm:**

From given word letter “G” = 071 and from the given keyword letter “A” = 065. From these two ascii values with mod the two. (71 % 26) = 19. (65 % 26) = 13. Add the two, 32. From this we mod (32 % 26) = 6. From this we then proceed with the vigenere table. And when we count from the given letter key “A” down 6 to “G” we then receive the first letter for the generated key which is letter “G”. This process is done repeatedly until the entire length of the user text is met.



**Program objectives:**

The objective of the program is to have the user provide a text that will be encrypted by a given keyword that the user also provides, and that encrypted word will also be decrypted for the user as well. The program does this job by providing a prompt for the user to enter text that will be encrypted with another prompt having the user enter the keyword they would like to use for the encryption. Once those two inputs are entered, the user will be provided the generated key for the text, the encrypted text, and to ensure it was done correctly, the decrypted text.

**Implementation of discrete structures:**

The knowledge gained from discrete structures were used in the program, such as the understanding of algorithms with the cipher using logical algebra to calculate the ascii values for the required encryption within the program. And discrete structures knowledge of functions with inclusion of understanding efficiency with a program was used to provide an efficient and accurate program. One example for a basis of understanding algorithms is the understanding the ascii values for converting a string input to a char input, that would be used to substitute another letter for the alphabet in that char input for the encryption.

**Limitations:**

The limitations that I had ran into when developing the program, was that of the inability to have the user enter a space (“ “) key into the provided text and to output the encrypted text with a space as well. The issue that I had kept running into, was the inability to calculate for another letter in the ascii value and to have that value outputted as a space input. I was unable to figure it out, and had to stick with having the user enter no spaces for the text. Another limitation was that the calculations I had provided were only able to calculate for capital letters, and I had attempted to solve a solution to this by inputting the user to a char array, but I was unable to solve that issue with only one character appearing. Therefore, it still stands that it only outputs encrypting used with capital letters.

**Improvements:**

Ways that I would improve the limitations of the program was to provide a solution for the ability for the user to input any text regardless of being capitalized and regardless of the space key, therefore it is a one to one encryption to the user’s text. Another way I would improve the program, is to have it accept number values within the conversion that would not interfere with the ascii value itself and cause the issue of going out of bounds.

**Pseudocode:**

Begin

Need 3 strings, keyword, user text, cipher

Ask user to enter text

Ask user to enter keyword to encrypt text

Text from user will be encrypted using keyword

Shift change the user text based on keyword

Change each character within the user string by converting it to a char array

Display encrypted text

For generating key same length as user text

Run forloop that will enter each value from keyword into a new string until it reaches the length of the user text

Decryption is done by reversing the process of encryption

Display encrypted, key, and decrypted text

End

**Flowchart:**

FOUND IN INCLUDED FILE