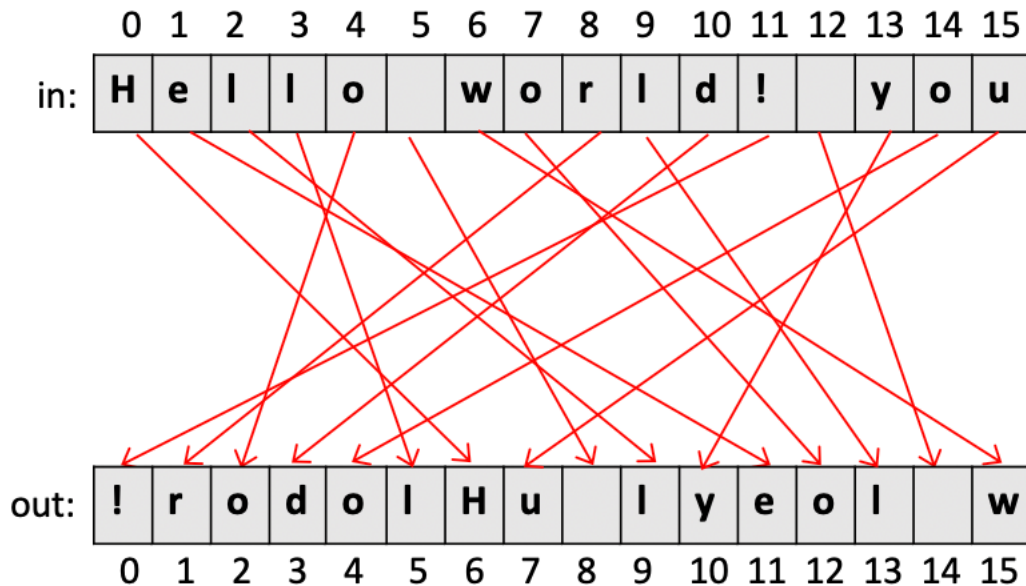


Learning Outcome: Students will gain experience in C programming using strings, loops, pointers, standard input, and functions.

Lab 3: The Permutation Cipher

Your task in this lab is to build a simple permutation cipher using C as follows:

1. Use the **fgets** function to prompt the user to enter plaintext.
2. If the user hits the enter key without entering any other characters, stop further execution.
3. If the length of the plaintext is less than 16, pad the plaintext using the underscore character ('_') to make it a string of length 16.
4. If the user types more than 16 characters, the rest of the input string will be ignored.
5. Encrypt the plaintext to obtain ciphertext according to the following permutation.



6. Print the ciphertext.
7. From the ciphertext, recover the original plaintext and print it.
8. Compare the recovered text and the plaintext and print the result.
9. Go back to step 1.

Additional Requirements:

- Use an appropriate data structure to represent the permutation rule and DRY (don't repeat yourself) like:

```
out[0] = in[11];
out[1] = in[8];
...
out[15] = in[6];
```

- Your main function shouldn't be too long and the following functions are required:
 1. padding
 2. encrypt
 3. decrypt
 4. main
- You may choose proper parameters and return type for each of the functions above.

Sample Output:

```
Enter a string: Hi there!
padded plaintext = Hi there!_____
ciphertext = !_h__tH_e _ie__r
recovered text = Hi there!_____
padded plaintext == recovered text? true
```

```
Enter a string: Once upon a time there was a lonely king
padded plaintext = Once upon a time
ciphertext = n ameOeucino tp
recovered text = Once upon a time
padded plaintext == recovered text? true
```

```
Enter a string: 0123456789ABCDEF
padded plaintext = 0123456789ABCDEF
ciphertext = B84AE30F52D179C6
recovered text = 0123456789ABCDEF
padded plaintext == recovered text? 1
```

```
Enter a string: 01234567890ABCDEF
padded plaintext = 01234567890ABCDE
ciphertext = A840D30E52C179B6
recovered text = 01234567890ABCDE
padded plaintext == recovered text? true
```

```
Enter a string: 0123456789ABCDE
padded plaintext = 0123456789ABCDE_
ciphertext = B84AE30_52D179C6
recovered text = 0123456789ABCDE_
padded plaintext == recovered text? true
```

Deliverables:

1. Create a directory named Lab3 under your assignment3 directory and place your code lab3.c (and other source files).
2. You must upload working code to your GitHub repository by 6 PM, 2/17/2021.
3. You must get sign-offs from the instructor or a TA during their office hours before the due date.

Sign offs – Each signature is worth $1/N$ of your lab grade where N is the number of signatures

- The student was able to accept user input and implement the padding function.
- The student was able to use arrays to implement the permutation rule and encrypt plaintext in a concise manner.
- The student was able to implement inverse permutation and decrypt ciphertext.
- The student was able to organize their C code using at least three functions.