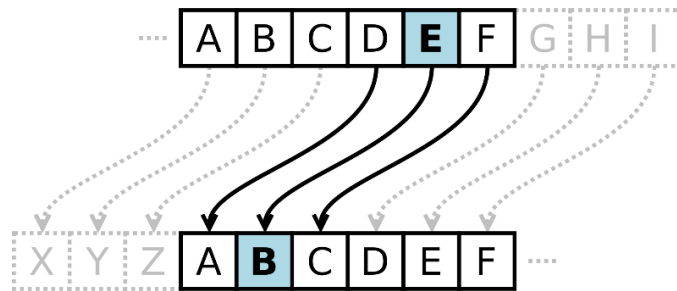


Veni Vidi Vici



In **cryptography**, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of **substitution cipher** in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.

For example, with a left shift of 3, D would be replaced by A, E would become B, and so on. The method is named after **Julius Caesar**, who used it in his private correspondence.



The encryption step performed by a Caesar cipher is often incorporated as part of more complex schemes, such as the Vigenère cipher, and still has modern application in the ROT13 system. As with all single-alphabet substitution ciphers, the Caesar cipher is easily broken and in modern practice offers essentially no communications security.

Cipher Text and Plain Text

In encryption, **plain text** is the normal, human-readable text. **Cipher text** is the encrypted message. In the header file, you'll find the two functions you are going to implement. One will encrypt plain text and the other will decrypt cipher text.

```
void cipher(istream& in, ostream& out, unsigned key);  
void plaintext(istream& in, ostream& out, unsigned key);
```

You are going to read from the parameter **in** and write to the parameter **out**. Treat the two parameters, **in** and **out**, just as you would **cin** and **cout**. In fact, if you want to encrypt the characters you type from the **keyboard**, and have the output appear in the

console, you can just call `cipher(cin, cout, 5)` from your run function and everything that you type will be shifted to the right by five characters.

Implementing the Functions

Both of these functions are **filter functions**, so they will use **data loops**. Here is the pseudocode for the `cipher()` function:

```
cipher(in, out, key):  
  read each character (c) from in  
    shift the character c right by key  
  write c to out
```

The plan for the `plaintext()` function is the same, except the character is shifted **to the left** by the **key** value. There are two important details:

- Only **alphabetic characters** (**a-z|A-Z**) are encrypted. Spaces and punctuation are written out unchanged.
- You will have to figure out how to **wrap around the characters** when shifting would cause it to move outside the alphabetic range. For instance, shifting the character **Z** right by 1 should convert it to **A**, and shifting the character **A** left by 1 should convert it to **Z**. You'll need to make use of the **%** operator to do this, along with some addition.

Once you've gotten 100% on all the tests go ahead and **submit** your code. If you have difficulty, ask on the discussion board or come to my office hours.