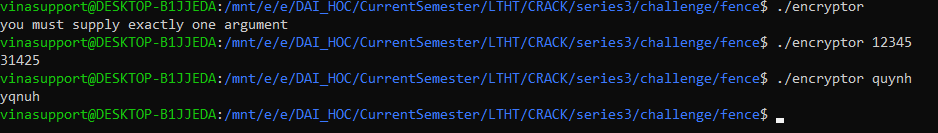
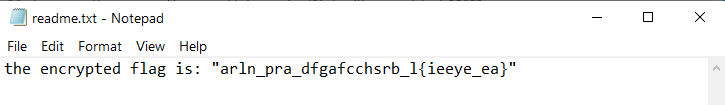
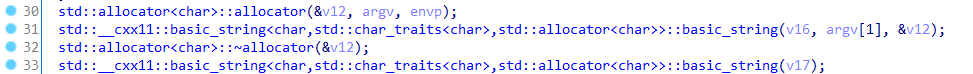
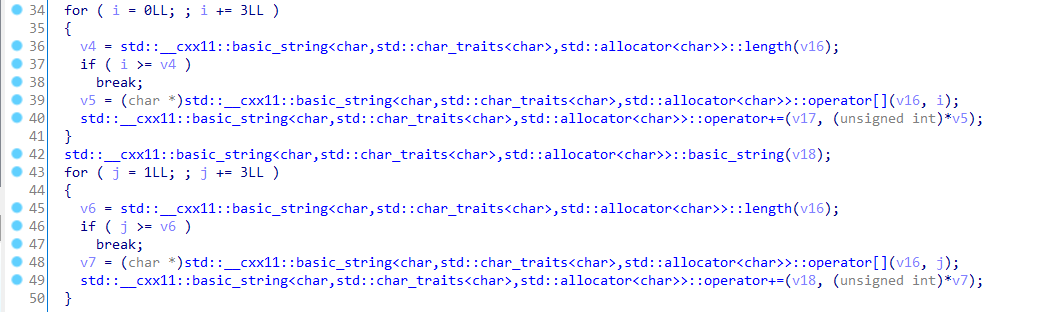
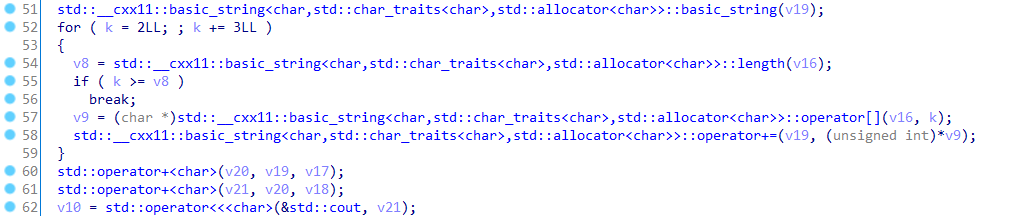
*Nguyễn Mỹ Quỳnh – 19520241*

**Fence**

Chạy thử chương trình, kết hợp với việc thấy tên chương trình là encryptor và đọc nội dung file readme.txt đã cho, ta có thể thấy: chương trình yêu cầu truyền vào một chuỗi plaintext và sau đó sẽ mã hóa và in ra ciphertext. Yêu cầu là với flag đã được mã hóa "arln\_pra\_dfgafcchsrb\_l{ieeye\_ea}" được cho trong readme.txt , cần tìm plaintext tương ứng.

Tiến hành đọc mã giả của hàm main cộng với việc suy đoán ta thấy:



+ v16 có thể là chuỗi plaintext mà ta nhập vào.

+ Vòng lặp từ dòng 34 đến 41, duyệt qua các kí tự trong chuỗi v16 vị trí thứ i, với i bắt đầu là 0 và bước nhảy là 3, với mỗi v16[i] được duyệt v17=v17+v16[i]. (dòng 40)

+ Vòng lặp từ dòng 43 đến 50, duyệt qua các kí tự trong chuỗi v16 vị trí thứ i, với i bắt đầu là 1 và bước nhảy là 3, với mỗi v16[i] được duyệt v18=v18+v16[i]. (dòng 49)

+ Vòng lặp từ dòng 34 đến 41, duyệt qua các kí tự trong chuỗi v16 vị trí thứ i, với i bắt đầu là 2 và bước nhảy là 3, với mỗi v16[i] được duyệt v19=v19+v16[i]. (dòng 58)

+ Dòng 60 đến 62 sẽ thực hiện tính v21= v19+v17+v18 và in ra v21 (chính là ciphertext).

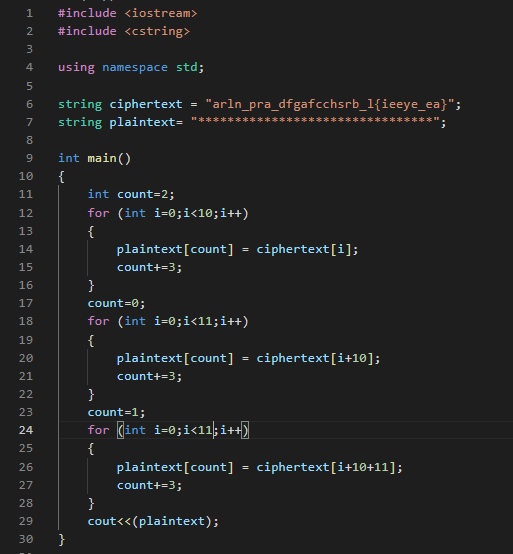
Với quy luật tính ciphertext từ plaintext như trên, áp dụng vào ciphertext "arln\_pra\_dfgafcchsrb\_l{ieeye\_ea}" ( 32 kí tự), vậy suy ra:

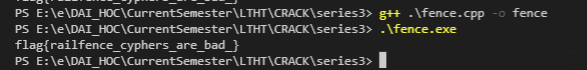
+ 10 kí tự đầu trong chuỗi ciphertext này sẽ là các kí tự thứ 2,5,8,…29 trong plaintext.

+ 11 kí tự tiếp theo trong chuỗi ciphertext này sẽ là các kí tự thứ 0,3,6,…30 trong plaintext.

+ 11 kí tự tiếp theo trong chuỗi ciphertext này sẽ là các kí tự thứ 1,4,7,…31 trong plaintext.

Dùng đoạn chương trình c++ sau để tìm plaintext



Ta tìm được plaintext là “flag{railfence\_cyphers\_are\_bad\_}”

Kiểm tra: