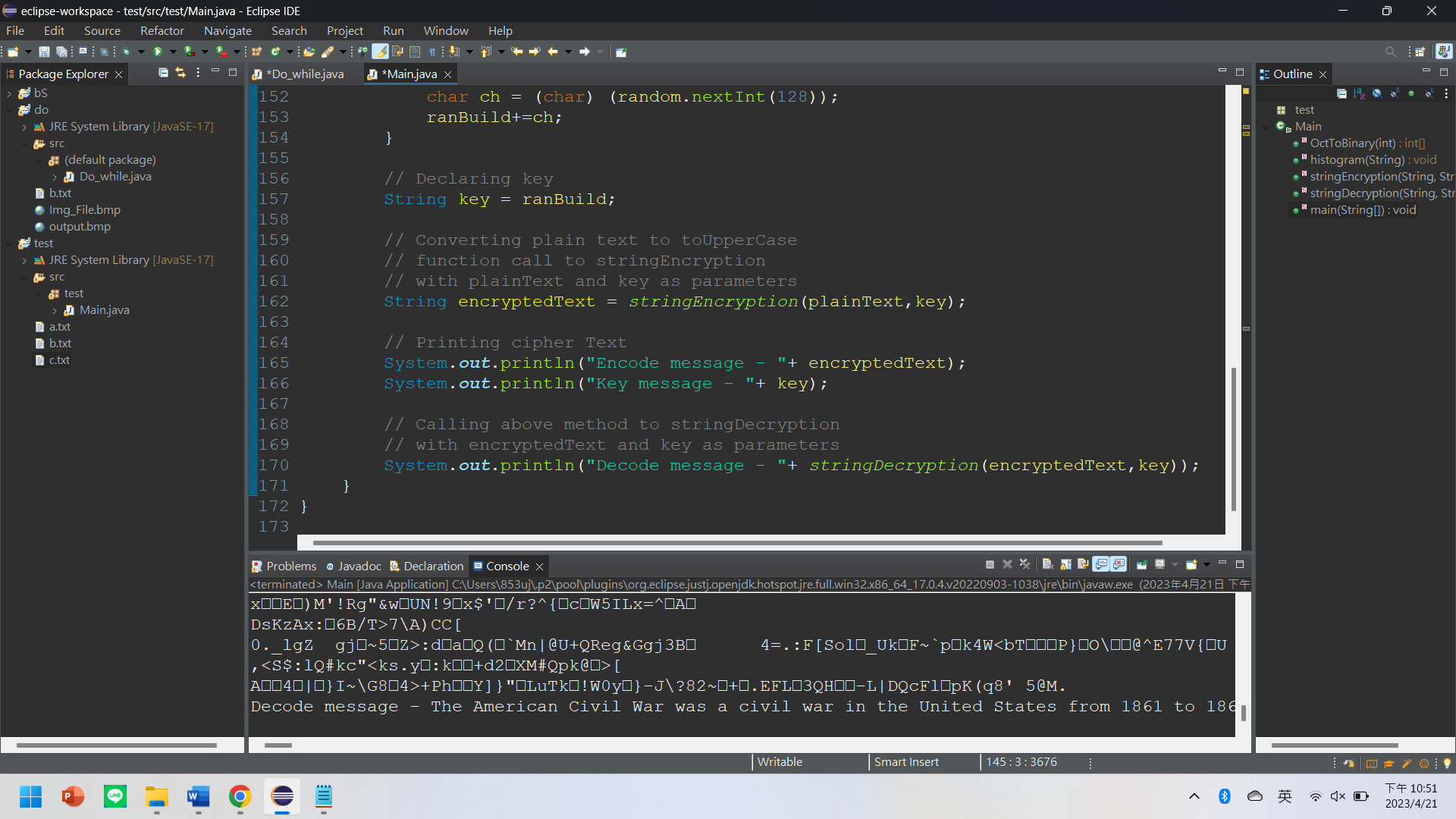
**版本1：將原文(a.txt)直接與隨機生成與原文相同長度的key做one time pad**



package test;

import java.util.\*;

import java.io.\*;

import java.nio.file.Files;

import java.nio.file.Paths;

public class Main {

public static int[] OctToBinary(int input) {

int[] arr = new int[8];

int[] arrRev = new int[8];

for(int i = 0; i<8 ; i++) {

arr[i] = 0;

}

int j = 0;

while(input!=0) {

if(input % 2 == 0) {

input = input/2;

j++;

}

else {

arr[j] = 1;

input = input/2;

j++;

}

}

int p=8;

for(int i=0;i<8;i++) {

arrRev[p-1]=arr[i];

p = p-1;

}

return arrRev;

}

// Method 1

// Returning encrypted text

public static String stringEncryption(String text,String key){

// Initializing cipherText

String cipherText = "";

// Initialize cipher array of key length

// which stores the sum of corresponding no.'s

// of plainText and key.

int cipher[] = new int[key.length()];

for (int i = 0; i < key.length(); i++) {

cipher[i] = text.charAt(i)

+ key.charAt(i);

}

// If the sum is greater than 25

// subtract 26 from it

// and store that resulting value

for (int i = 0; i < key.length(); i++) {

if (cipher[i] > 127) {

cipher[i] = cipher[i] - 128;

}

}

// Converting the no.'s into integers

// Convert these integers to corresponding

// characters and add them up to cipherText

for (int i = 0; i < key.length(); i++) {

int x = cipher[i];

cipherText += (char)x;

}

// Returning the cipherText

return cipherText;

}

// Method 2

// Returning plain text

public static String stringDecryption(String s,String key){

// Initializing plain text

String plainText = "";

// Initializing integer array of key length

// which stores difference

// of corresponding no.'s of

// each character of cipherText and key

int plain[] = new int[key.length()];

// Running for loop for each character

// subtracting and storing in the array

for (int i = 0; i < key.length(); i++) {

plain[i]

= s.charAt(i)

- key.charAt(i);

}

// If the difference is less than 0

// add 26 and store it in the array.

for (int i = 0; i < key.length(); i++) {

if (plain[i] < 0) {

plain[i] = plain[i] + 128;

}

}

// Converting int to corresponding char

// add them up to plainText

for (int i = 0; i < key.length(); i++) {

int x = plain[i];

plainText += (char)x;

}

// Returning plainText

return plainText;

}

public static void main(String[] args){

//Load a.txt

File file = new File("a.txt");

String data = "";

Scanner sc;

try {

sc = new Scanner(file);

while (sc.hasNext()) {

// Check if there is more input to be read

data = sc.nextLine();

}

sc.close();

} catch (FileNotFoundException e) {

System.out.print("Stop");

}

// Declaring plain text

String plainText = data;

Random random = new Random();

String ranBuild = "";

for (int i = 0; i < data.length(); i++) {

char ch = (char) (random.nextInt(128));

ranBuild+=ch;

}

// Declaring key

String key = ranBuild;

// Converting plain text to toUpperCase

// function call to stringEncryption

// with plainText and key as parameters

String encryptedText = stringEncryption(plainText,key);

// Printing cipher Text

System.out.println("Encode message - "+ encryptedText);

System.out.println("Key message - "+ key);

// Calling above method to stringDecryption

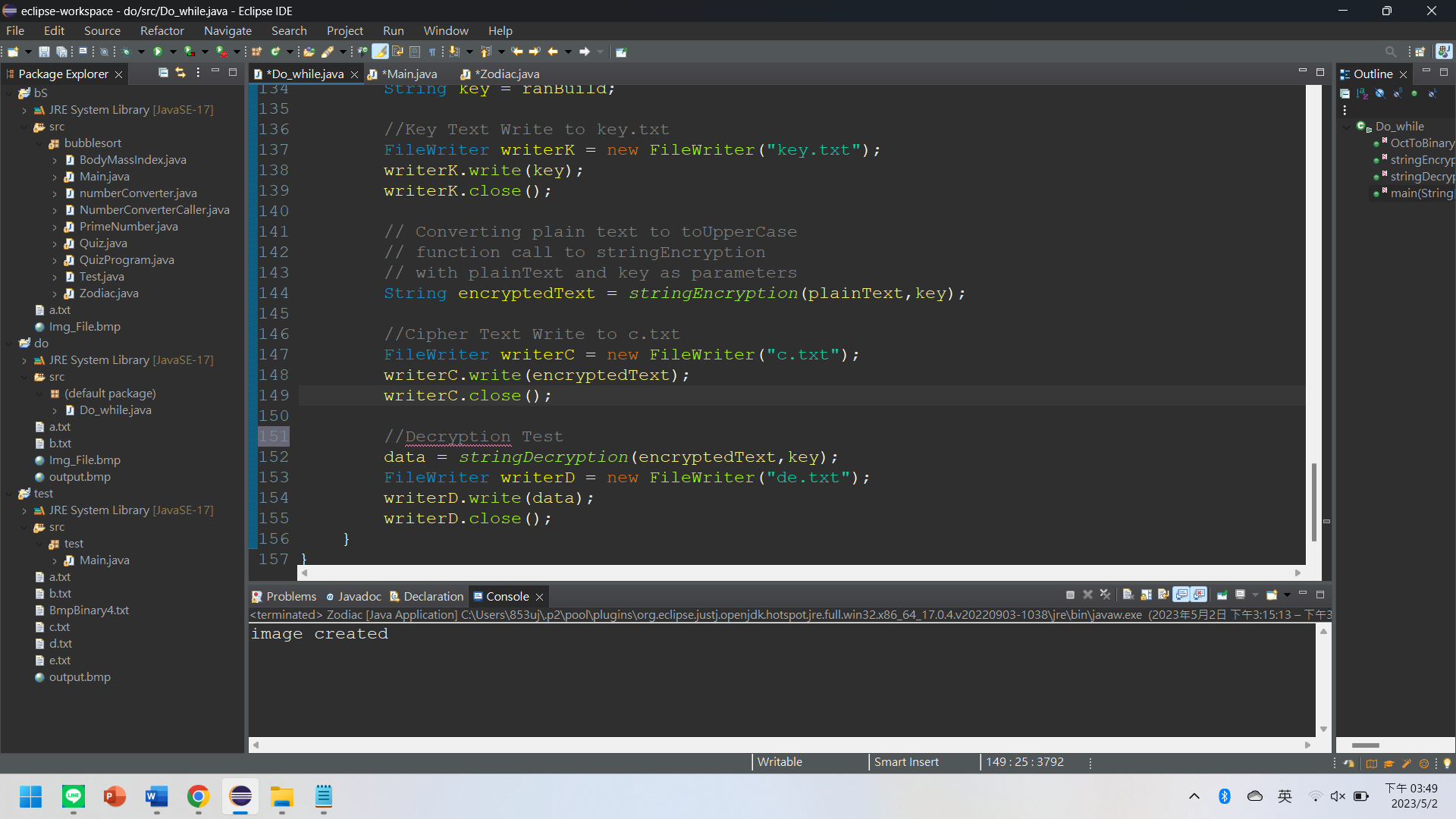
// with encryptedText and key as parameters

System.out.println("Decode message - "+ stringDecryption(encryptedText,key));

}

}

**版本2：將密文寫入c.txt 鑰匙寫入key.txt 轉回的原文寫入de.txt**



import java.util.\*;

import java.io.\*;

public class Do\_while {

public static int[] OctToBinary(int input) {

int[] arr = new int[8];

int[] arrRev = new int[8];

for(int i = 0; i<8 ; i++) {

arr[i] = 0;

}

int j = 0;

while(input!=0) {

if(input % 2 == 0) {

input = input/2;

j++;

}

else {

arr[j] = 1;

input = input/2;

j++;

}

}

int p=8;

for(int i=0;i<8;i++) {

arrRev[p-1]=arr[i];

p = p-1;

}

return arrRev;

}

// Method 1

// Returning encrypted text

public static String stringEncryption(String text,String key){

// Initializing cipherText

String cipherText = "";

// Initialize cipher array of key length

// which stores the sum of corresponding no.'s

// of plainText and key.

int cipher[] = new int[key.length()];

for (int i = 0; i < key.length(); i++) {

cipher[i] = text.charAt(i)

+ key.charAt(i);

}

// If the sum is greater than 127

// subtract 128 from it

// and store that resulting value

for (int i = 0; i < key.length(); i++) {

if (cipher[i] > 127) {

cipher[i] = cipher[i] - 128;

}

}

// Converting the no.'s into integers

// Convert these integers to corresponding

// characters and add them up to cipherText

for (int i = 0; i < key.length(); i++) {

int x = cipher[i];

cipherText += (char)x;

}

// Returning the cipherText

return cipherText;

}

// Method 2

// Returning plain text

public static String stringDecryption(String s,String key){

// Initializing plain text

String plainText = "";

// Initializing integer array of key length

// which stores difference

// of corresponding no.'s of

// each character of cipherText and key

int plain[] = new int[key.length()];

// Running for loop for each character

// subtracting and storing in the array

for (int i = 0; i < key.length(); i++) {

plain[i]

= s.charAt(i)

- key.charAt(i);

}

// If the difference is less than 0

// add 128 and store it in the array.

for (int i = 0; i < key.length(); i++) {

if (plain[i] < 0) {

plain[i] = plain[i] + 128;

}

}

// Converting int to corresponding char

// add them up to plainText

for (int i = 0; i < key.length(); i++) {

int x = plain[i];

plainText += (char)x;

}

// Returning plainText

return plainText;

}

public static void main(String[] args) throws IOException{

//Load a.txt

File file = new File("a.txt");

String data = "";

Scanner sc;

try {

sc = new Scanner(file);

while (sc.hasNext()) {

// Check if there is more input to be read

data = sc.nextLine();

}

sc.close();

} catch (FileNotFoundException e) {

System.out.print("Stop");

}

// Declaring plain text

String plainText = data;

Random random = new Random();

String ranBuild = "";

for (int i = 0; i < data.length(); i++) {

char ch = (char) (random.nextInt(128));

ranBuild+=ch;

}

// Declaring key

String key = ranBuild;

//Key Text Write to key.txt

FileWriter writerK = new FileWriter("key.txt");

writerK.write(key);

writerK.close();

// Converting plain text to toUpperCase

// function call to stringEncryption

// with plainText and key as parameters

String encryptedText = stringEncryption(plainText,key);

//Cipher Text Write to c.txt

FileWriter writerC = new FileWriter("c.txt");

writerC.write(encryptedText);

writerC.close();

//Decryption Test

data = stringDecryption(encryptedText,key);

FileWriter writerD = new FileWriter("de.txt");

writerD.write(data);

writerD.close();

}

}