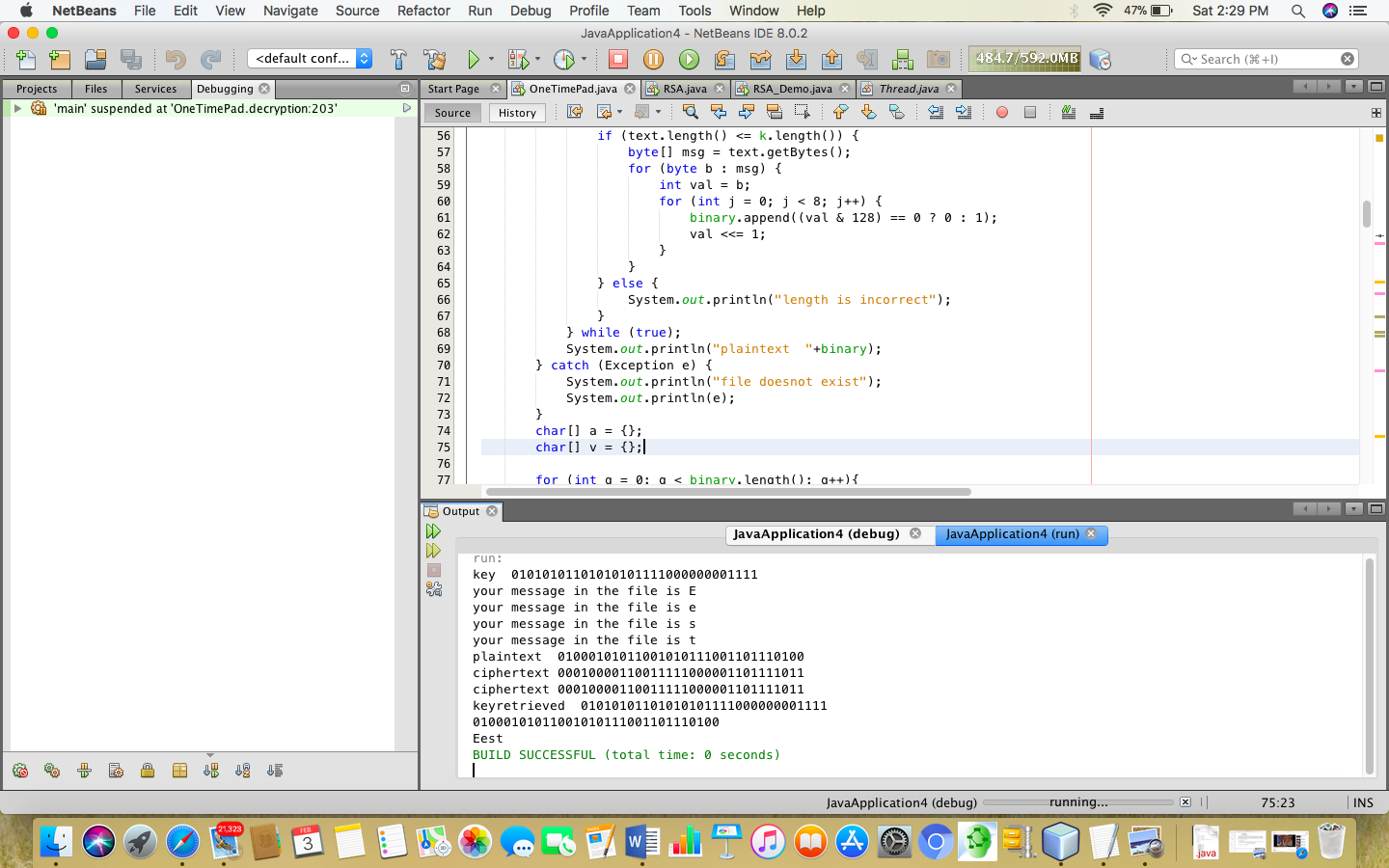
**Report 1**

1. **Program Description and working of functions:**

The result of the program after implementing functions of encryption and decryption the results are



In this screenshot data is retrieved from file plaintext.txt which is eest and then key is retrieved from key.txt.

Plaintext is converted into binary string of data and exclusive or is implemented on it along with secret key (Similar to the key given in an example in the question).

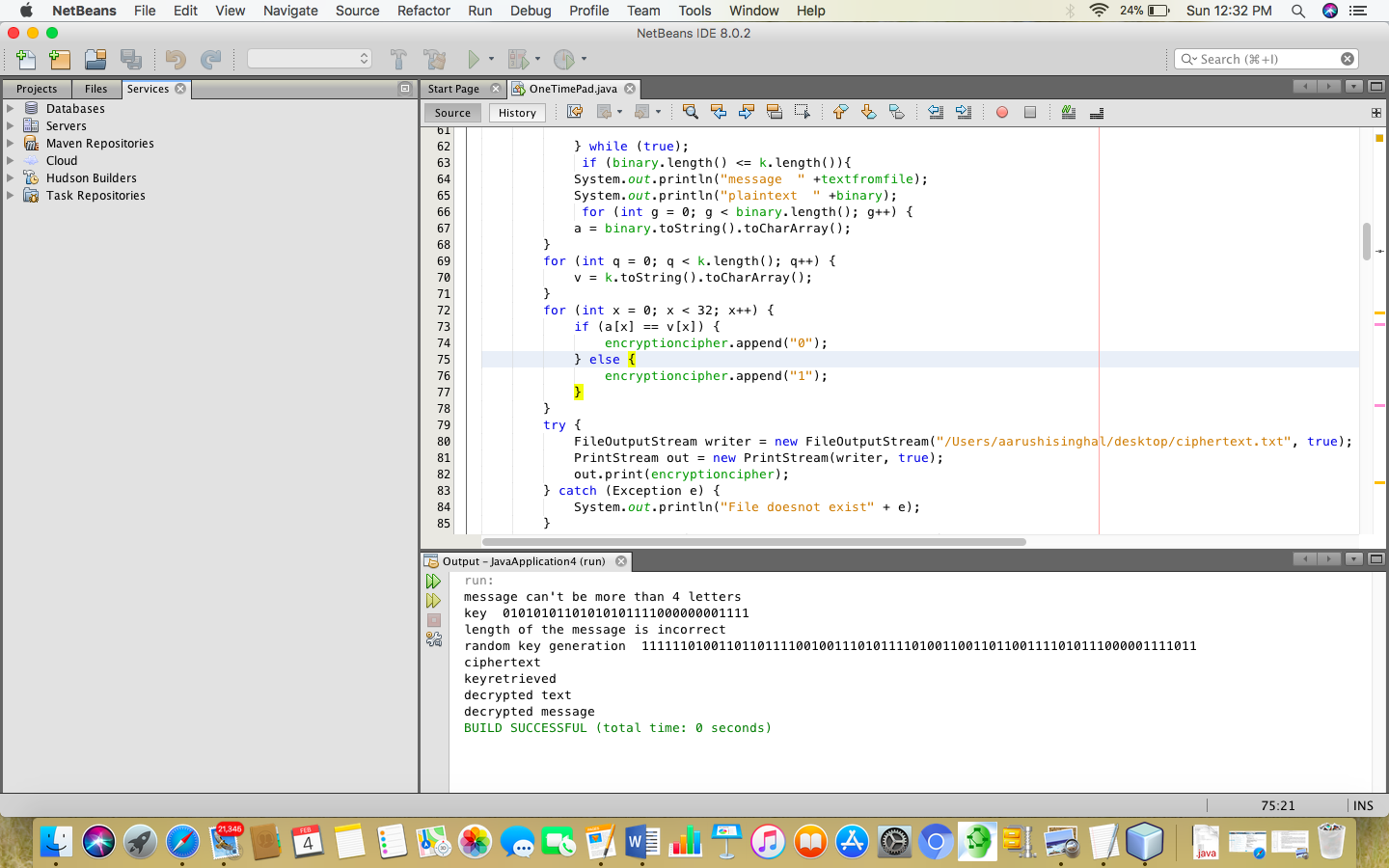
The result after implementation of exclusive or is stored in ciphertext.txt and is also shown as ciphertext to check the validity of data. This generation of ciphertext takes places in enc () function.

In decryption function, data is read from ciphertext which is produced in enc () function and key is read from key.txt, then to retrieve plaintext again exclusive or is performed between ciphertext and key and result is stored in result.txt after the binary data is converted into plaintext which was actually the message. The message is also displayed when you run the code as decrypted message.

The message cannot be more than key so,

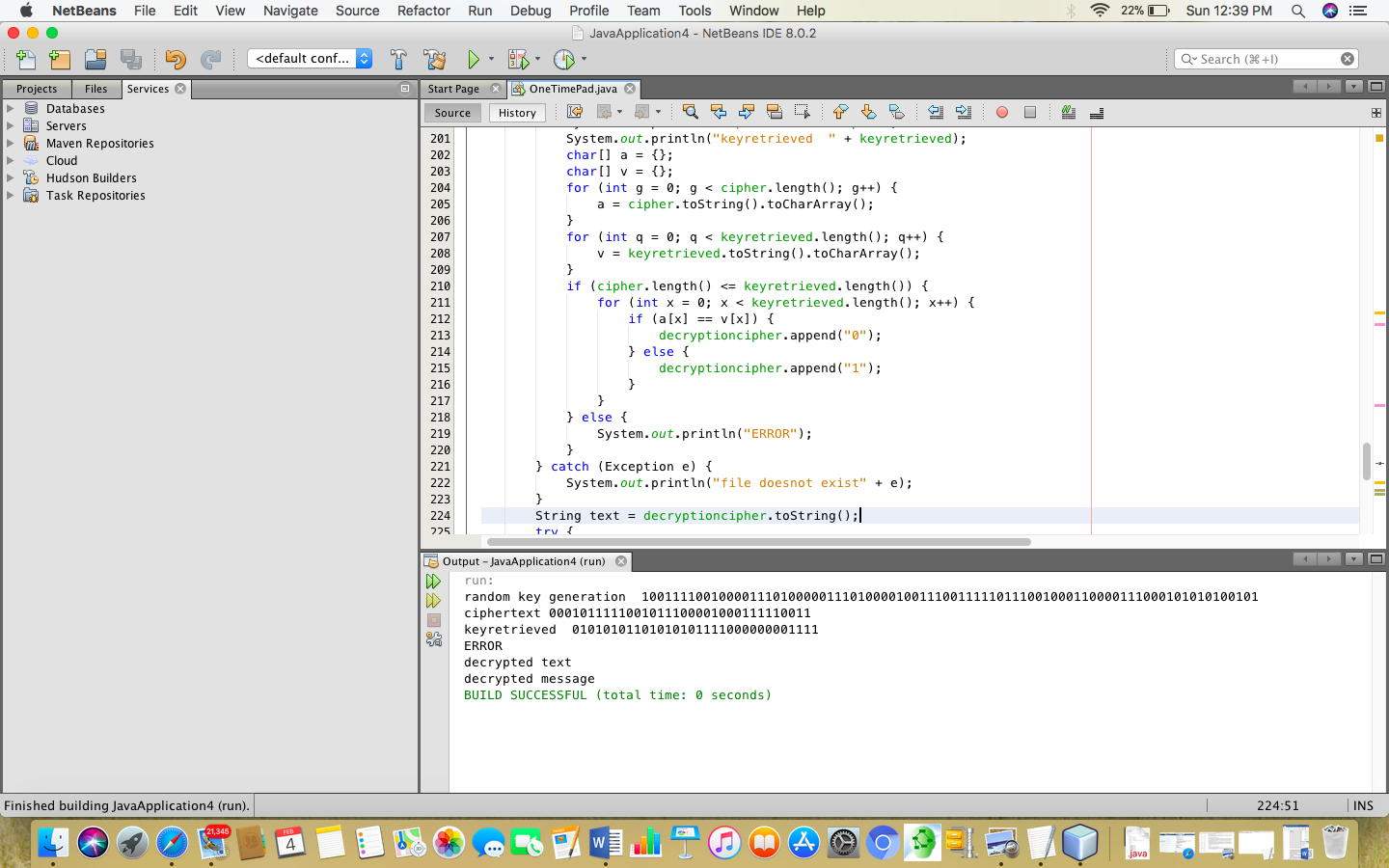
Key is 32 bits so message can also be of 4 letters not more than 4 letters.

If length is more than 4 which more than the size of key the message will show length is incorrect.



In the above screen shot it can be clearly seen if message is more than 4 letters or size of key the message will pop saying that “length of the message is incorrect”.

Similarly, is the case with decryption () function in which if the size of ciphertext is greater than the secret key then the message saying “ERROR” is popped up.



This how encryption and decryption function works.

1. **Frequencychecker () function to calculate frequency of key and proof of uniform distribution**

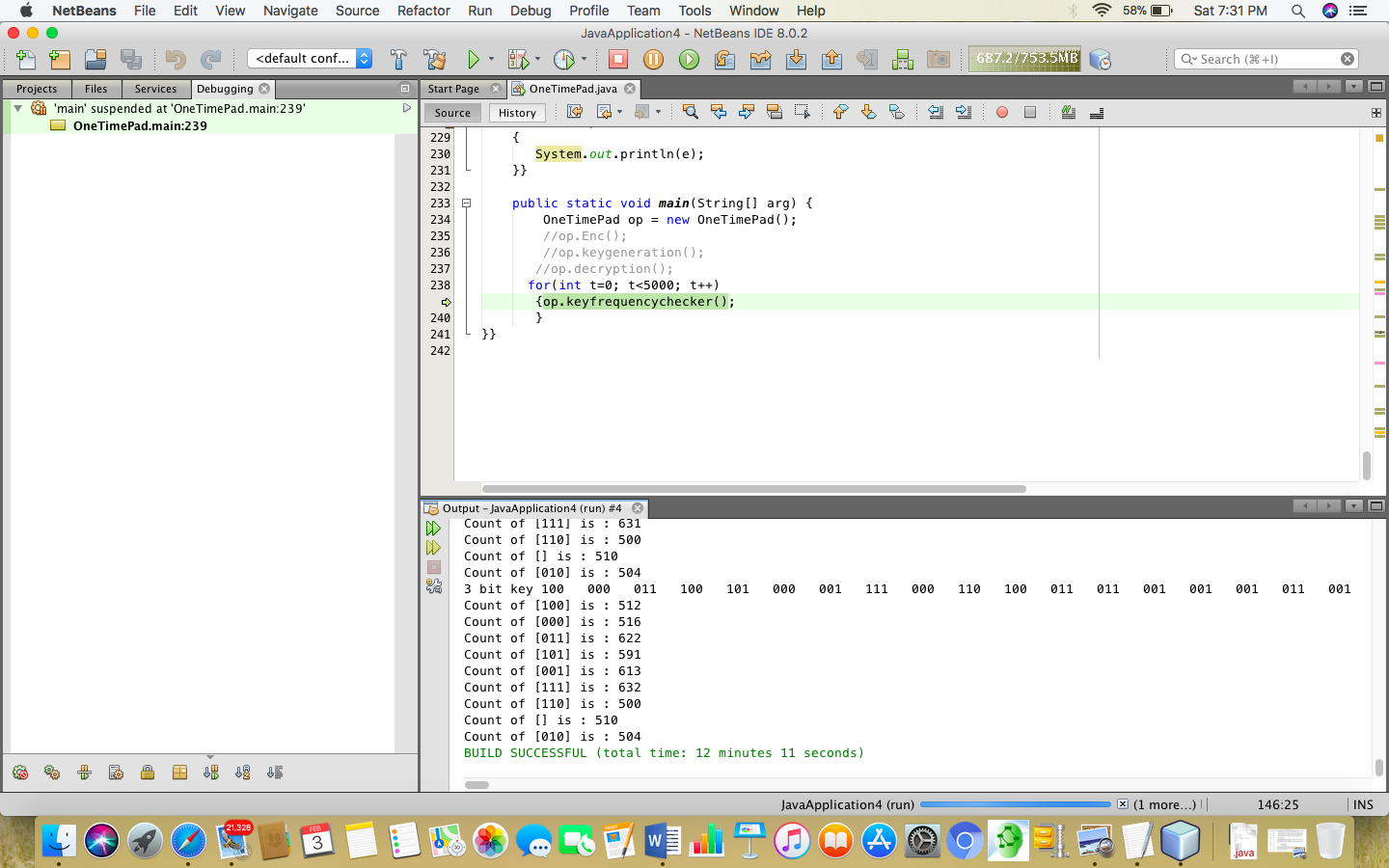
The function frequencychecker () is the function meant to check the security key of length 3 bits length.

As the key is binary so there are only 2 inputs 0 or 1

{0,1} ^n where n is 3 so, number of possible combinations are 8 which are as follows

000, 001, 010, 011, 100, 101, 110, 111

This function after being called in the main class calculates the number of same substring that is being generated during the process of 5000 counts and increments count every time it encounters that particular string of binary combination.



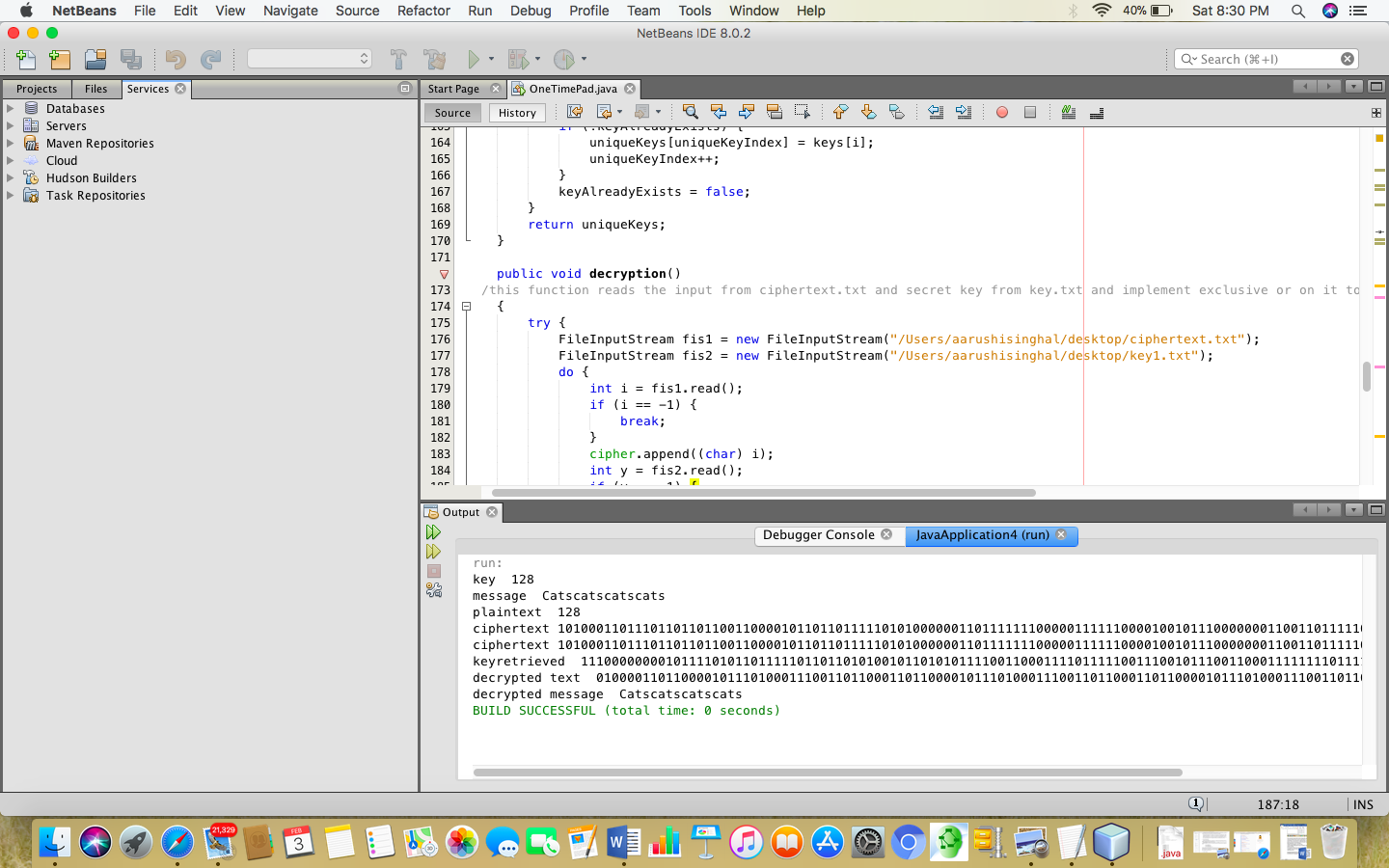
This screenshot shows that function along with the output which shows the frequency of each substring repeated when the key is generated for 5000 times. The frequency of each combination is near 550 which is approximately very close to each other that shows the key is almost uniformly distributed as all the keys get equal chance of being generated.

Yet it is proved by the chat as it shows uniform distribution.

|  |  |  |
| --- | --- | --- |
| S. no | Binary combination | Frequency |
| 1 | 000 | 516 |
| 2 | 101 | 591 |
| 3 | 110 | 500 |
| 4 | 111 | 632 |
| 5 | 001 | 613 |
| 6 | 011 | 622 |
| 7 | 010 | 504 |
| 8 | 100 | 512 |

**Average Running Time for encryption**

Encrypting text using enc () function with 128 bits key take around 179776677151511 nanoseconds on an average. So, it can be concluded that as the key size increases so does the time will also increase to encrypt the data

****

**3.The configuration**

Operating System : Mac Ox

Programming language is : Java (IDE 8.0.2).

Crypto packages being used are “Secure Random” to generate secure random numbers being stored in the Random keyGenerator () function and frequencychecker () to check that key is uniformly distributed or not.

In encryption function two inputs are there which the function takes from two files that are message and the secret key which are used to generate ciphertext which gets stored in ciphertext.txt file.

java version "1.8.0\_144"

Java(TM) SE Runtime Environment (build 1.8.0\_144-b01)

Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode