**Project Title**: SDES Encryption Decryption with Java

**Group Members:**

* Anaesthasia Patricia (001201900111)
* Muhamad Afif Fadillah (001201900010)
* Farel Arden (001201900068)

1. **Project Overview**

Our product function is to translate the original text to be the cipher-text or vice versa.  
However, we have already declared the input, here are the inputs:

* 10-bit key: 0101010101
* 8-bit plaintext: 10101010
* 8-bit ciphertext: 00101010

In this project, we want to use the OOP concept, therefore in this project, we have 2 classes and 1 interface instead of functions, the only function that we have is turning the integers into decimals to get the rows and columns on both S0 and S1.

1. **The Purpose**

Before we do encryption or decryption, we produced two 8-bit subkeys from a 10-bit key. Two 8-bit subkeys used in particular stages of encryption or decryption.

S-DES (Simplified-Data Encryption Standard) encryption processed the original-text (8-bit block) into the cipher-text (8-bit block) and vice versa for S-DES (Simplified-Data Encryption Standard) decryption.

1. **Scope of  Work**

Included:

* 2 Class and 1 Interface
* IP (Initial Permutation) Process
* Fk Process (E/P, S0 and S1, P4) with K1
* SW (Switch Function) Process
* Fk Process (E/P, S0 and S1, P4) with K2
* IP-1 (Inverse Permutation) Process

Not included: 8-bit plain text and 10-bit key input

1. **System Requirements**

We no have specific system requirements to develop or run our program.

We need Java’s IDE to develop this program. We use Eclipse IDE for Java Development and Java Development Kit (JDK). For users can run our program in Java’s IDE too.

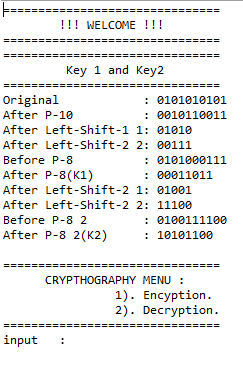
1. **User Manual**

**5.1 How to Install**

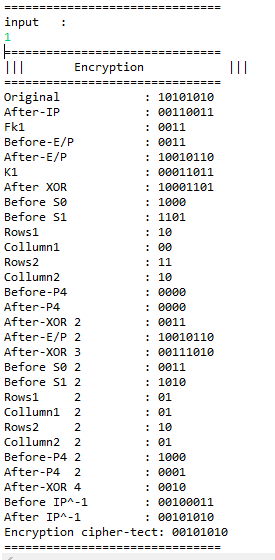
Link: <https://drive.google.com/open?id=1wnxzUdeNXv5KlDA2RrF0R04c-mAshdLw>

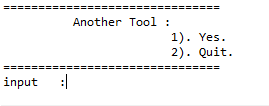
1. After opening the link, users double click the zip file. Download the zip file.
2. Open Eclipse IDE for Java Development. After opening it, click File> New> Java Project. Enter a project name with “encryption” and click Finish.
3. Click the “encryption” package in Package Explorer. Click the right mouse to “encryption” and choose New> Package. Create a new Java package with an “encryption” name and click Finish.
4. Click the right mouse to “encryption” package and choose New> Class. Create a new Java class with the “CryptoApp” name and click Finish.
5. Open the zip file. After that, double click the encryption file. Double click to open the “CryptoApp.java” file and copy the code.
6. Delete “package encryption; public class CryptoApp {}” and paste the copied code.
7. For “Crypto.java”, click right mouse to “encryption” package and choose New> Interface. Create a new Java interface with the “Crypto” name and click Finish.
8. Back to WinRAR, double click to open the “Crypto.java” file and copy the code.
9. Repeat the 6th step.
10. For “BasicCrypto.java”, repeat the 4th until 6th steps.
11. Click Run> Select All> Ok. The program is run.

**5.2 How to Use**

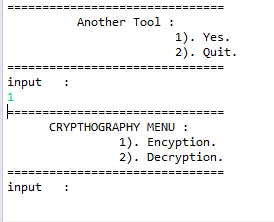


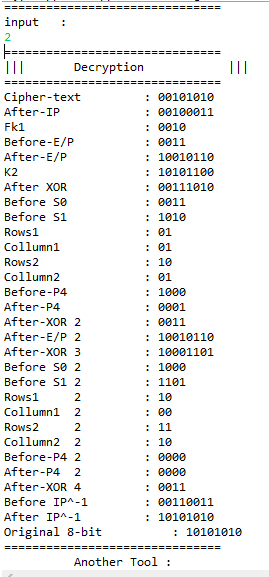
This is the “Welcome” screen. In this screen, we show the process to produce two 8-bit subkeys from a 10-bit key. Below that, we show the cryptography menu. Users can input 1 for encryption or 2 for decryption. If users input 1, it will be show encryption like below.



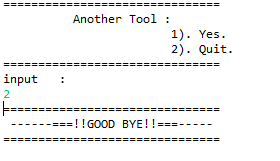


For “Another Tool”, it shows like below if users input 1.





There is a decryption in above. For “Another Tool” if users input 2, it will be shown like this and the program ends.



1. **Conclusion**

This program is run successfully and met its goals. We will fix several typos and we make with GUI if we have given more time and resources.