Instructor Name:	Dr. Osama Amjad
Student Name:	Aye Kyi Kyi Cho
Student ID:	1276026

Contents

DES Encryption and Avalanche Effect GUI Application 1
Features 1
Requirements 2
Code Overview 2
Using Inputs to Run a Test 2
Output Screenshots 3

DES Encryption and Avalanche Effect GUI Application

This Python application provides a graphical user interface (GUI) to perform DES (Data Encryption Standard) encryption and analyze the Avalanche Effect in DES. The program includes two main tabs:

a) DES Encryption with detailed round-by-round output
b) Avalanche Effect comparison between two plaintexts differing by one bit

---

Features

- Full DES encryption of 64-bit plaintext using a 64-bit key
- Displays detailed DES rounds: round number, subkey (Ki), left half (Li), and right half (Ri) for each round
- Calculates and visualizes the Avalanche Effect by comparing intermediate round outputs of two similar plaintexts
- User-friendly input validation with clear error messages
- Nicely formatted tabular display of all results using the tabulate library
- GUI built with tkinter and ttk for a native desktop look and feel

---

Requirements

- Python 3.x
- tkinter (usually included in Python standard library)
- tabulate library (for pretty printing tables)

Install tabulate via: **pip install tabulate**

Code Overview

Core Functionalities

- Utility functions for hex/binary conversions, permutations, XOR, and shifts
- Standard DES tables (initial permutation, inverse, expansion, S-boxes, etc.)
- Key scheduling to generate 16 subkeys from a 64-bit key
- Feistel function implementation for each DES round
- Complete encryption flow with initial and final permutations
- Avalanche effect calculation to compare two plaintexts differing by one bit

GUI

- Built with tkinter and ttk for a tabbed interface
- Input validation with error messages for incorrect inputs
- Scrollable text output areas for displaying results
- Buttons for running encryption and avalanche effect analyses

Hardcoded Constants vs. Default Inputs

- The DES algorithm's core tables — such as **Initial Permutation (IP)**, **Inverse Initial Permutation (IP^{-1})**, **Expansion (E)**, **Permutation (P)**, **S-boxes**, and key scheduling tables — are **hardcoded constants** embedded in the code. These are standard and essential parts of DES and do not change during program execution.
- Separately, the application provides **default hardcoded input values** for the user input fields (plaintexts and keys) in the GUI tabs. These defaults serve to simplify demonstration and testing, allowing users to run the program immediately without entering data manually.

Using Inputs to Run a Test

DES Encryption Tab (a)

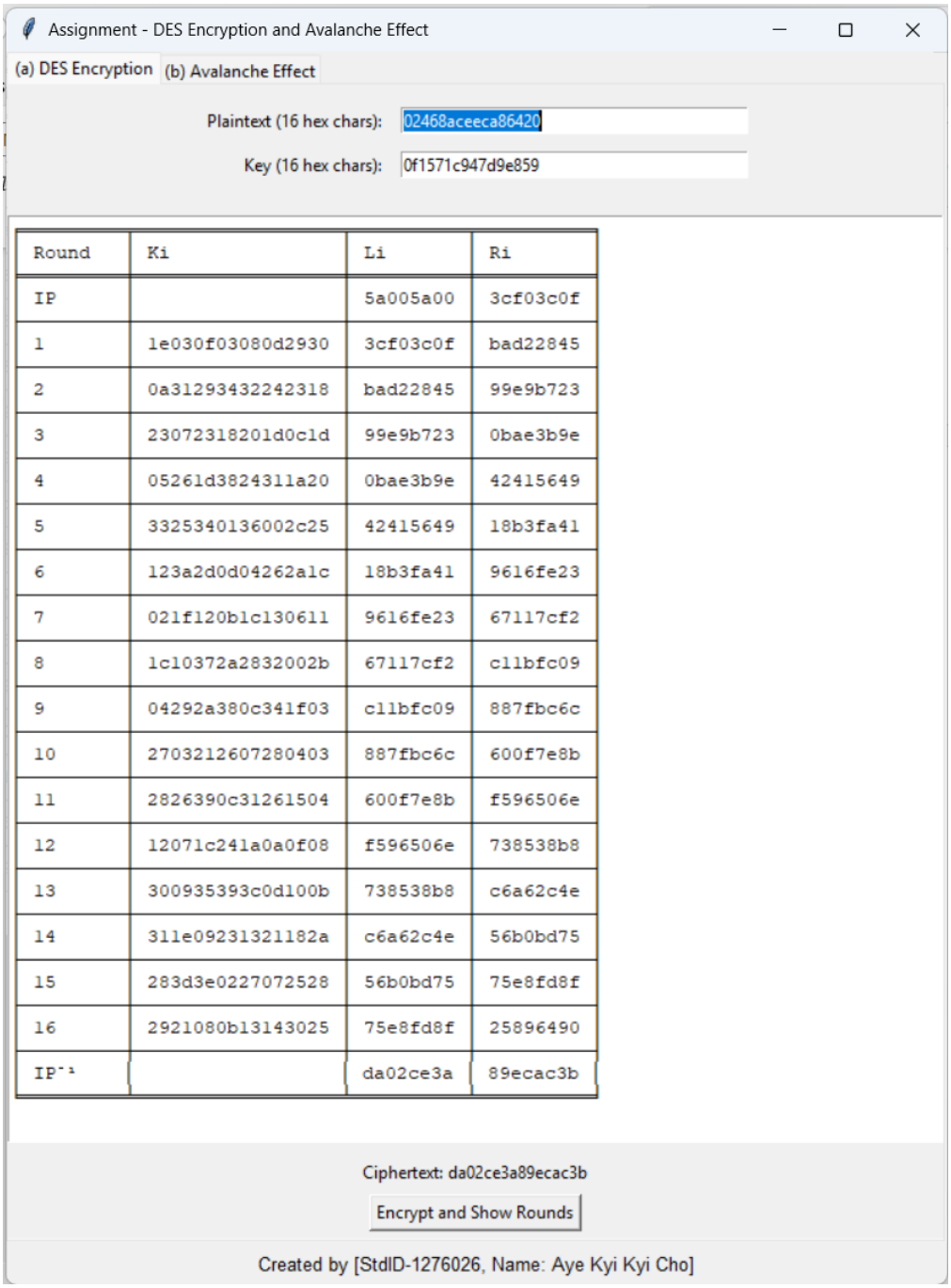
- Plaintext (16 hex chars): 02468aceeca86420
- Key (16 hex chars): 0f1571c947d9e859
- Click "Encrypt and Show Rounds"

Avalanche Effect Tab (b)

- Plaintext 1 (16 hex chars): 02468aceeca86420
- Plaintext 2 (16 hex chars): 12468aceeca86420
- Key (16 hex chars): 0f1571c947d9e859
- Click “Show Avalanche Effect”

Output Screenshots

(a) DES Encryption Tab Result



Round	Ki	Li	Ri
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP^{-1}		da02ce3a	89ecac3b

(b) Avalanche Effect Tab Result