**Ministry of Education, Culture and Research of Republic of Moldova**

**Technical University of Moldova**

**Faculty of Computers, Informatics and Microelectronic**

**Department of Software Engineering and Automatics**

**Cryptography and Security**

***Laboratory Work 4:Block ciphers. The DES algorithm***

Elaborated:

st. gr. FAF-211 Gazea Sandu

Verified:

asist. univ. Cătălin Mîțu

Chişinău - 2023

**Task:**

To develop a program in one of the programming languages preferred for implementing an element of the DES algorithm. The task will be chosen according to the order number n of the student from the group list, according to the formula: nr\_task = n mode 11. For each task, the tables used and all intermediate steps should be displayed on the screen. Input data may be user-entered or randomly generated.

2.1. Given the key of the DES algorithm (8 symbols), to determine K+.

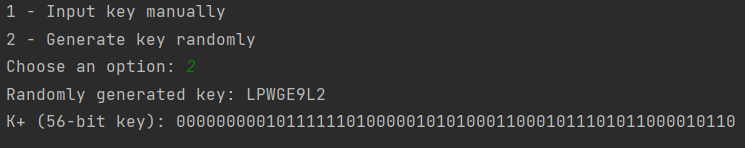
**Implementation**

The main() function orchestrates the overall program flow. It first invokes get\_key() to obtain the key from the user or generate it randomly. The key is then converted into a 64-bit binary representation, where each character is represented by its ASCII value in binary, padded to 8 bits. The binary key is processed through the apply\_PC1 function to produce the 56-bit permuted key, key\_56\_bit, commonly referred to as "K+" in DES terminology. Finally, it prints the "K+" key to the console.

* **apply\_PC1(key\_64\_bit)**: This function takes a 64-bit binary string and applies the PC1 table to permute the key according to the DES algorithm, resulting in a 56-bit key.
* **generate\_random\_key()**: Generates a random 8-character key using uppercase letters and digits. This key will be used as the input for the DES algorithm if the user chooses to generate a key randomly.
* **get\_key()**: This function presents the user with the menu to either input a key manually or generate one randomly. It ensures that the manually entered key is exactly 8 characters in length.

The script's user interface is simple and interactive, requiring input from the user to determine the flow of execution. It's designed to be a demonstration of the initial key processing step in the DES algorithm rather than a full implementation of DES. The output includes the permuted 56-bit key, which is a crucial component in the DES key schedule where it is later split and shifted to generate 16 round keys used in the encryption process.

**Output:**



***Task 2.1***

**Conclusions**

The provided script elegantly encapsulates the initial permutation phase of the Data Encryption Standard (DES), a critical early step in the algorithm’s elaborate process to secure information. This implementation not only makes the intricacies of DES accessible but also serves as a testament to the algorithm's significance in the history of cryptography. Despite DES's vulnerability to today's advanced brute-force attacks, it remains an important subject of study, highlighting essential concepts in the field of encryption. The simplicity and clarity of the script's approach to demonstrating the permutation of the DES key further underscore the profound impact that DES has had on setting the standards for secure communication. Its design principles have informed the development of stronger, more secure encryption algorithms that are vital in safeguarding digital data against the backdrop of ever-increasing computational power.

Github: *https://github.com/GSandu1/CS.git*