Deadlines:

For all Groups: December 13 2022

Grading system

- 1 problem - 6

- 2 problems - 7

- 3 problems - 8

- 4 problems - 9

- 5 problems and bonus– 10

1. Random vulnerability

You've stumbled onto a significant vulnerability in a commonly used cryptographic library. It turns out that the random number generator it uses frequently produces the same primes when it is generating keys.

Exploit this knowledge to factor the (hexadecimal) keys below, and enter your answer as the last six digits of the largest factor you find (in decimal).

Key 1:

```
1c7bb1ae67670f7e6769b515c174414278e16c27e95b43a789099a1c7d55c717b2f0a0442a7d4
9503ee09552588ed9bb6eda4af738a02fb31576d78ff72b2499b347e49fef1028182f158182a0b
a504902996ea161311fe62b86e6ccb02a9307d932f7fa94cde410619927677f94c571ea39c7f41
05fae00415dd7d
```

Key 2:

```
2710e45014ed7d2550aac9887cc18b6858b978c2409e86f80bad4b59ebcbd90ed18790fc56f53f
fabc0e4a021da2e906072404a8b3c5555f64f279a21ebb60655e4d61f4a18be9ad389d8ff05b99
4bb4c194d8803537ac6cd9f708e0dd12d1857554e41c9cbef98f61c5751b796e5b37d338f5d9b
3ec3202b37a32f
```

2. RSA

Make a program that will implement the RSA algorithm made by you and with help of this program you can encrypt any string and then decrypt it.

Restrictions:

You must not use any libraries for this exercise.

3. Check password

We all use many passwords every day but is every password we use hard enough to crack. Your password is considered strong if it follows this rules:

-It has at least 8 characters and at most 20 characters.

-It has a minimum of 1 lower case letter [a-z] and a minimum of 1 upper case letter [A-Z] and a minimum of 1 numeric character [0-9] and a minimum of 1 special character: ~`!@#$%^&*()-_+={}[]|\;:"<>,./?

-It does not contain three repeating characters in a row (i.e.,“abbbcc-0”is weak, but is “ abcbbcc-0”strong).

Given a string password, return *the minimum number of steps required to make the password strong. If the password is already strong, return the message “good”.*

In one step, you can:

-Insert one character

-Delete one character

-Replace one character

Restrictions:

You must not use any libraries for this exercise.

Example 1:

Input: password = "a"

Output: 7

Example 2:

Input: password = "Bb1"

Output: 5

Example 3:

Input: password = "aaB-d2c1"

Output: good

4. Palindrome

You are given a string **str**. You can convert **str** to a palindrome by adding characters in front of it. Return the shortest palindrome you can find by performing this transformation.

Restrictions:

You must not use any libraries for this exercise.

Example 1:

Input: str = "aabbcbbaaa"

Output: “aaabbcbbaaa”

Example 2:

Input: str = "abcd"

Output: “dcbabcd”

5. Cracking the password

You obtained a secret message that was encrypted using Vigenère cipher with a simple pass phrase. Thus, it is vulnerable to frequency analysis. Decode this message and find out what does this message say and what is the encryption key.

Cipher message:

OOGNVMTNTCLUOGZSZSHTXAZGMOMEPKWDDQM

Bonus: Mastermind

In this problem we need to create our own Mastermind game. In our version of Mastermind game users can choose between two versions of the game: auto and manual. In the auto version the computer will randomly select a secret code and the user tries to guess it, based on the deterministic clues given by the computer and in the manual version the user will input the secret code and another user tries to guess it based on clues that appear on screen. You must make this game playable in the terminal.

Bonus task:

Make for your game a graphical interface