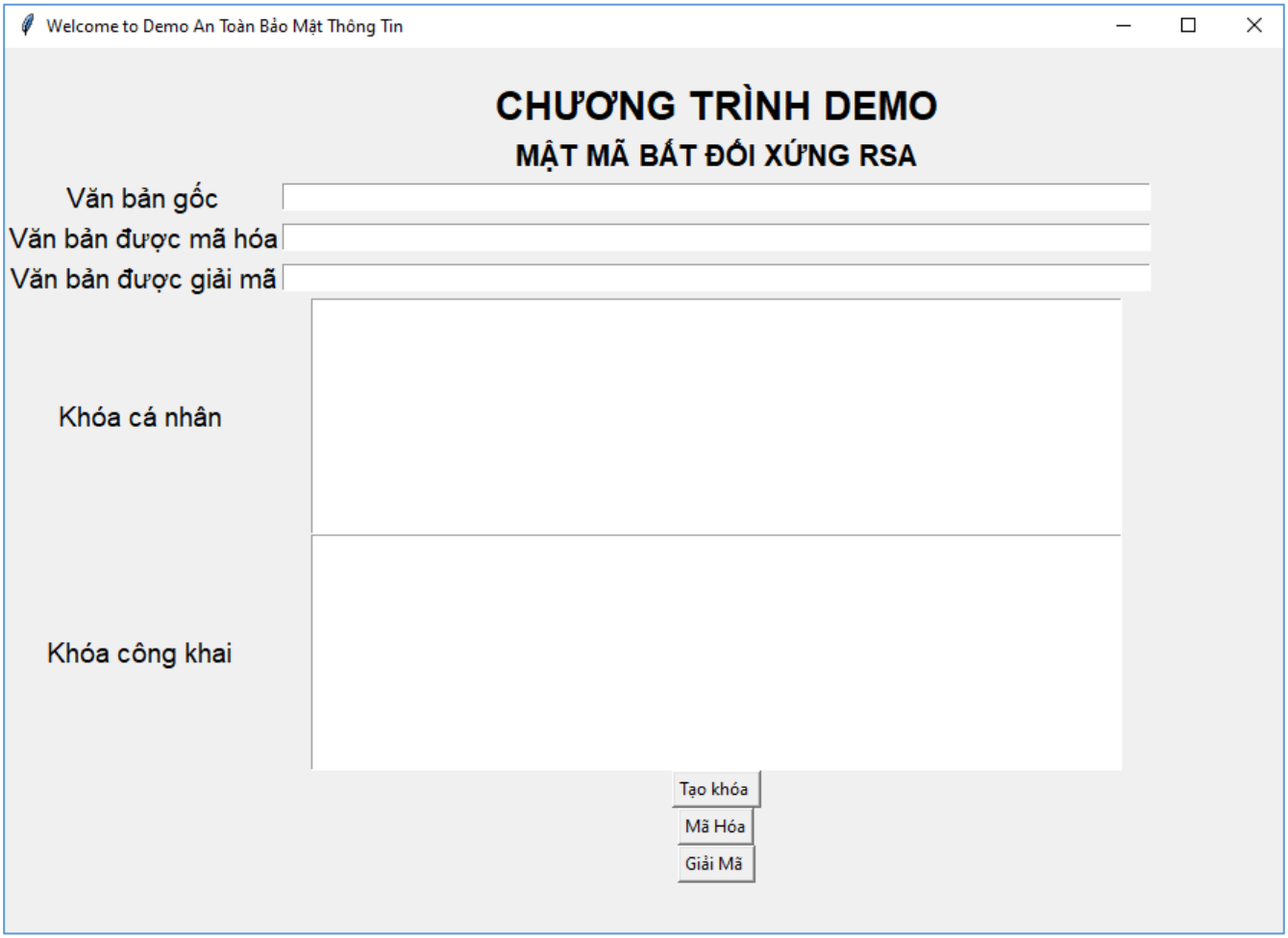
1. Bài 1.

Bước 1. Hãy tạo giao diện như sau (học viên có thể sử dụng thư viện tkinter, Qt, Flask, ...)



(Gợi ý; xem lại bài thực hành buổi 1)

Bước 2. Cài đặt sự kiện nhấn chuột cho nút Tạo Mã. Cho phép người sử dụng lưu lại tập tin khóa công khai và khóa riêng tư

Gợi ý:

```
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Hash import SHA
from Crypto.Cipher import PKCS1_v1_5
import base64
from tkinter import *
from tkinter import filedialog
```

```
def generate_key():
    key = RSA.generate(1024)
    pri = save_file(key.exportKey('PEM'),
                    'wb',
                    'Lưu khóa cá nhân',
                    (("All files", "*.*"), ("PEM files", "*.pem")),
                    ".pem")
    pub = save_file(key.publickey().exportKey('PEM'),
                    'wb',
                    'Lưu khóa công khai',
                    (("All files", "*.*"),("PEM files", "*.pem")),
                    ".pem")
    pri_key.delete('1.0',END)
    pri_key.insert(END,key.exportKey('PEM'))
    pub_key.delete('1.0',END)
    pub_key.insert(END,key.publickey().exportKey('PEM'))
```

Gợi ý cài đặt thủ tục **save_file**

```
def save_file(content, _mode, _title, _filetypes,
_defaultextension):
    f = filedialog.asksaveasfile(mode = _mode,
                                 initialdir = "C:/",
                                 title = _title,
                                 filetypes = _filetypes,
                                 defaultextension = _defaultextension)
    if f is None: return
    f.write(content)
    f.close()
```

Bước 3. Cài đặt sự kiện nhấn chuột cho nút Mã Hóa, sử dụng khóa công khai đã được lưu ở bước 2 để mã hóa văn bản gốc bằng giải thuật RSA

Gợi ý:

```
def get_key(key_style):
    filename = filedialog.askopenfilename(initialdir = "C:/",
                    title = "Open " + key_style,
                    filetypes = (("PEM files", "*.pem"),("All
files", "*.*")))
    if filename is None: return
    file = open(filename,"rb")
    key = file.read()
    file.close()
    return RSA.importKey(key)
```

```
def mahoa_rsa():
    txt = plaintxt.get().encode()
    pub_key = get_key("Public Key")
    cipher = PKCS1_v1_5.new(pub_key)
    entxt = cipher.encrypt(txt)
    entxt = base64.b64encode(entxt)
    ciphertxt.delete(0,END)
    ciphertxt.insert(INSERT,entxt)
```

Ghi chú:

- Cần cài đặt thư viện pycryptodome (https://pypi.org/project/pycryptodome/) nếu sử dụng Python3.
- Hoặc thư viện pycrypto (https://pypi.org/project/pycrypto/) nếu sử dụng Python2. Trước đó cần cài Microsoft Visual C++ Compiler for Python 2.7 (https://www.microsoft.com/en-gb/download/details.aspx?id=44266)

Bước 4. Học viên tự cài đặt sự kiện cho nút Giải mã.

Gợi ý: sử dụng phương thức giải mã cipher.decrypt

2. Bài 2. Không sử dụng thư viện mã hóa RSA. Hãy viết chương trình mã hóa, và giải mã theo các bước của giải thuật.

Bước 1: Viết hàm sinh khóa

- Sinh 2 số nguyên tố lớn
- Tìm USCLN(e, φ(n))
- Tìm nghịch đảo của e theo mod(φ(n))
- ……..

- Chọn 2 số nguyên tố lớn p và q
- Tính n = p * q
- Tính φ(n) = (p-1) * (q-1)
- Chọn e sao cho USCLN(e, φ(n)) = 1 với 1 < e < φ(n)
- Tính d sao cho ed ≡ 1 (mod φ(n))

  {e,n} – public key

  {d,n} – private key

Bước 2: Viết chương trình mã hóa và giải mã

Giải mã:

Học viên nộp bài (tập tin Python) trên hệ thống elcit. Buổi thực hành này được điểm danh bằng các bài nộp trên hệ thống Elearning .

Ở đầu file python cần ghi rõ các thông tin sau:

```
# Họ và tên sinh viên:
# Mã số sinh viên:
# STT:
```