전자서명 숙제

1. 4장 강의 노트 15~16페이지의 ECDSA 알고리즘을 파이썬으로 구현한다. 타원 곡선은 아래와 같은 SECP256K1 타원 곡선을 이용한다.

p = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F

e1 = (0x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798,
0x483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8)

q = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364141

전자서명 프로그램은 sign() 함수와 verify() 함수로 구성되는데, sign(M, d) 함수는 메시지 M과 개인키 d를 인자로 받아서 전자서명 S1과 S2를 생성하여 return 한다.
verify(M, S1, S2, e2) 함수는 메시지와 전자서명 S1, S2, 그리고 공개키 e2를 인자로 받은 후, 메시지의 내용과 전자서명이 일치하면 True를 반환하고, 아니면 False를 return 한다. 이때 프로그램의 검증을 위해 A와 B의 내용을 출력한다.

프로그램이 정상적으로 실행하는지를 검사하기 위해서 아래와 같은 코드를 추가한다.

```
if __name__ == "__main__":
    d = ec.generate_private_key()        # 2주차 과제에서 작성한 함수
    e2 = ec.generate_public_key(d)       # 2주차 과제에서 작성한 함수

    M = input("메시지? ")
    S1, S2 = sign(M, d)
    print("1. Sign:")
    print("\tS1 =", hex(S1))
    print("\tS2 =", hex(S2))

    print("2. 정확한 서명을 입력할 경우:")
    if verify(M, S1, S2, e2) == True:
        print("검증 성공")
    else:
        print("검증 실패")

    print("3. 잘못된 서명을 입력할 경우:")
    if verify(M, S1-1, S2-1, e2) == True:
        print("검증 성공")
    else:
        print("검증 실패")
```

프로그램 검증을 위하여 d의 값을 3으로 가정했을 경우, 실행 결과의 예는 다음과 같다.

```
메시지? Secret message
1. Sign:
        S1 = 0xf06d76e6364b9e31f36c18c4174fdd21c8aac52a7f238ba362bff718b477e70a
        S2 = 0xd46899ff4dd7be8636911a9ae29e20c3c5acb5fbd3c24c3a208e4b981a820e64
2. 정확한 서명을 입력할 경우:
        A = 0xc87c6b04f148478ee1cfbd9a85da08a957a62739a0b2464521c7091da0bd0800
        B = 0xdb6392e67548df457633bcb1383326961ec94443115886389b3df8b0636e1e45
검증 성공
3. 잘못된 서명을 입력할 경우:
        A = 0x5190d36185bbc0cfa8d77037881def28eac50ab8bfda3247bdc63c79cf3aa092
        B = 0x3702a33ede1fd843b620380fe744c1788b9ec2bff4fea754bd0d3006cdb2ef15
검증 실패
```