Smart Contract 숙제

1. 경매를 진행하는 Smart contract의 첫 번째 version을 구현하자. 기본적으로 다음과 같은 함수들이 구현되어야 한다.
   - 생성자 : owner가 실행하며, 희망 가격(uint)과 경매 기간(uint)를 parameter로 전달한다. 희망 가격을 입찰하는 사람이 있으면 경매는 즉시 종료된다. 혹은 입찰 시간이 경매 기간을 초과할 경우에도 경매는 종료된다. 경매 기간은 파이썬의 int(time.time()) 값을 사용하여 계산한다.
   - bid() : 입찰자가 msg.value에 입찰 가격을 입력하여 실행한다. 경매가 종료되었거나 입찰 가격이 현재까지의 최대 가격에 미달하면, 실행을 취소한다. 아니면 입찰 가격을 최대 가격에 등록하고, 입찰자도 기록한다. 이전의 최대 입찰 가격을 제시한 입찰자에게는 입찰 금액을 즉시 전송한다. 최대 입찰 가격이 바뀔 때마다, 입찰자와 입찰 가격을 event에 기록하라.
   - finish() : owner가 실행하며, 입찰 종료 조건이 만족될 때만 실행된다. 최종 입찰 가격과 입찰자를 event에 등록하고, 입찰 금액을 owner에게 전송한다.

2. 위의 Smart contract는 bid()에서 심각한 문제가 존재한다. 즉, 이전의 입찰자에게 입찰 금액을 전송할 때, 전송받는 입찰자가 contract가 될 수 있고 receive() 함수에서 악의적인 코드가 실행될 수 있다. 이를 해결하기 위하여, smart contract의 두 번째 version을 아래와 같이 구현하라.
   - 생성자와 finish() : 동일
   - bid() : (주소, 반환 금액)의 mapping을 사용하여, 이전 입찰자에게 반환할 금액의 누계를 저장한다.
   - withdraw() : 입찰자가 반환받을 금액이 있는지 확인한 후, 그 금액을 전송한다. mapping 정보도 수정해야 하는데, 반드시 “금액을 전송하기 전에 mapping 정보를 수정”해야 한다. 왜 그렇게 해야하는지 생각해보라.

1번 프로그램을 auction1.sol, 2번 프로그램을 auction2.sol에 저장하여 제출한다.