Exercise #3: **Market**

Details:

This contract implements a simple marketplace with a simple escrow system. In case of a dispute the ether are stuck in the contract.
Anyone can create an offer to sell something by calling the addOffer function with the name of the product and the price in wei. The function then returns the id of the offer. This id should also be the position of the offer in the array returned by the getter function offers. The offers function returns an array of all offers past and present. After creation the offer is in the OFFERED state and an addOffer event is emitted.

Anyone can take an offer that is currently in the OFFERED state by calling the takeOffer function with the id and sending the right amount of ether. If the id is invalid or the sent ether does not match the price the message fails. Otherwise the offer goes into the TAKEN state, the sender of the message becomes the taker and a OfferTaken event is emitted.

The taker of an Offer can confirm an offer. This should happen after the shipment has been received. The ether paid for the offer is then forwarded to the creator of the offer. The offer is then in the CONFIRMED state and a OfferConfirmed Event is emitted.

If the taker does not confirm, the ether remains stuck in the contract.

Interface:

```
enum Status { OFFERED, TAKEN, CONFIRMED}
struct Offer { /* FIELDS GO HERE */ };
function addOffer(string product_, uint price_) returns (uint id);
function takeOffer(uint id);
function confirm(uint id);
function offers() returns (Offer[]);
event OfferAdded(uint indexed id, string indexed product, uint indexed price);
event OfferTaken(uint indexed id);
event OfferConfirmed(uint indexed id);
```

Example:

Somebody creates the contract.
Carol takes offer #5. The message fails, there is no such offer.
Carol confirms offer #5. The message fails, there is no such offer.
Alice adds an offer “3 DAO tokens” for 10 ether. The call returns id 0.
Carol confirms offer #0. Nothing happens, Carol is not the taker of #0.
Bob adds an offer “2 ether” for 1 ether. The call returns id 1.
Carol takes offer #0 and sends 2 ether. Nothing happens, the ether don’t match the price.
Bob takes offer #0 and sends 10 ether. The contract now has 10 ether. Bob is the taker.
Carol takes offer #0 and sends 10 ether.They are returned, the offer is already taken.
(Alice sends 3 DAO token to Bob.) Bob confirms #0 and Alice gets 10 ether.
The contract now has 0 ether.
Bob confirms #0. Nothing happens, the offer was already confirmed.
Carol takes offer #1 and sends 1 ether. Carol is the taker.
(It’s a scam!) Carol never calls confirm. The 1 ether is stuck in the contract forever.

Relevant global variables: msg.sender, msg.value
Hints: array.push(value), auto-generate the offers function