Level 6 - Verifying Signatures

For this level we signed some messages off chain using the following front end code:

```
const ethers = require('ethers');

let messageHash = ethers.utils.id("bidPrice(0.420)");
let messageHashBytes = ethers.utils.arrayify(messageHash);
let flatSig = await wallet.signMessage(messageHashBytes); // Sign the binary data
let sig = ethers.utils.splitSignature(flatSig);  // sig.v sig.r sig.s etc
```

Using the **Isolution6** interface write a function that will take the messageHash (plus params) and return the signer of the message.

Hint: Don't forget to prepend your message with ***“\x19Ethereum Signed Message:\n32”***

```
interface Isolution6 {
    function solution(
      bytes32 messageHash,
      uint8 v,
      bytes32 r,
      bytes32 s
      ) external pure
    returns (address signer);
}
```