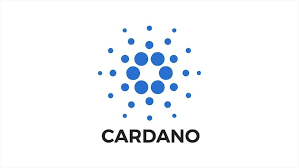
**SPONSOR CHALLENGE**



**Cardano/IOHK Challenges**

This is the Marlowe Challenge for the Hackathon

An option gives its owner the opportunity to buy something at a given point in the future. Typically, it allows someone to fix a (lower, or predictable) price for something that might become more expensive. This challenge asks you to use Marlowe – either embedded in Haskell or using Blockly – to build a suite of options contracts. These should include as many of the following as you can, but you may also want to think of implementing different extensions or combinations of these. \* Simple option contract: pay 100 Ada at the start for an option at the end time (e.g slot 10). The item purchased is not modeled within the contract.

\* Extend 1 to make a multi-stage option contract: payments for an option at slot 20 can be paid at times 0 and 10.

\* An option contract for a coupon bond. In this case, the item optioned is modeled as a part of the contract.

\* A two-participant option contract. On an option for something valued at 100 Ada, each participant can pay 30-70 Ada: if the first person wants to pay N Ada, then the second person pays the difference between that payment and 100.

\* Add a guarantor to the option: they guarantee to buy the optioned item even if the option holder does not.

\* Build a double option, in which A takes an option on X, and B on Y; before realizing the options, participants are given a chance to swap X and Y, settling the difference in costs for the swap to be successful.

**ACTUS challenge:** The ACTUS foundation has provided a taxonomy and specification of a wide range of financial contracts. This challenge is to provide the most comprehensive, well-coded and documented set of ACTUS contracts in Marlowe. The judges will particularly look for a solution which gives templates (or functions) from which a range of contacts can be instantiated. They will also look for good coding style, under which common patterns have been abstracted out into library functions.

**Marlowe innovation challenge:** Marlowe is designed to help people to write financial contracts on blockchain, but it can also be used for other kinds of contracts. The prize for this challenge will be awarded for the most innovative and novel use of Marlowe.

**Marlowe extension challenge.** How would you extend Marlowe to make it easier to write financial contracts on blockchain? The prize for this challenge will be awarded to the most fully realized extension to Marlowe. This could, for example, comprise a set of additional constructs for Marlowe, for additional features on top of Marlowe, or for changes to the implementation.

*Things To Keep In Mind*

* Good use of Haskell libraries/tooling in Plutus contract
* Code is good but not required.
* Probably some code should be required.

We would suggest e Haskell, otherwise we don't mind. The goal is to integrate with the Plutus technology preview, and not anything else.

**Prizes**

Plutus Challenge - $2,000

Marlowe Challenge - $2,000

Options Challenge - $2,000

**Judges**

Anantha, Karthik, Peter