#Instructions

- Please make a copy before you edit it: File -> Make a copy.
- Please find the problem statement and detailed template below.
- From where the template starts you will be allowed only 3 pages for the solution summary
- Please submit the final solution document with an access link in the submission form

Girl Hackathon

[Do not edit this section. This is read-only]

Problem Statement:

As discussed in workshops, manufactured chips may have structural faults at certain places/nodes, which must be tested before being delivered to end users.

The task is to design an algorithm and write its code to identify the input vector required to identify the fault at a given node in a given circuit.
In a case, there would only be a single fault in the design.

The algorithm should be efficient, robust and able to identify faults quickly.

Inputs

Available inputs are -

1. Circuit file (format provided below)
2. Fault node location
3. Fault type
    a. SA0 : stuck-at-0, a fault where node is not able to attain value 1, irrespective of inputs
    b. SA1 : stuck-at-1, a fault where node is not able to attain value 0, irrespective of inputs

Outputs

The code should print a vector for inputs to test the fault, and the expected value of output to confirm the fault.
The output should be printed to the following file in run directory - output.txt

Circuit Format

1. The circuit will have 4 inputs - A, B, C and D. All of which are boolean type (only 0 and 1 are valid inputs)
2. The circuit's output will always be Z which is also a boolean.
3. The circuit will be built using the following operations -
    a. AND (&) gate
    b. OR (|) gate
    c. NOT (~) gate
    d. XOR (^) gate
4. The circuit would purely be a combinational logic.
5. All internal nodes in the circuit would be named as : "net_<alphanumeric string>"
6. Each input (A / B / C / D) would be utilized only once in the circuit.

Example Input:

Circuit File:

```
net_e = A & B
net_f = C | D
net_g = ~ net_f
Z = net_g ^ net_e
```

Fault:

```
FAULT_AT = net_f
FAULT_TYPE = SA0
```

Example Output:

[A, B, C, D] = [0, 0, 0, 1], Z = 1

Participants need to work on the above problem statement and provide the solution for the same. Goodluck!

Submission:

Participants are required to create a PDF document as the final submission. The document should contain the link to a public GitHub repository (accessible and open to all).

The repository should have all the collaterals of the code, along with a README file. The code can be written in any open-source programming language using standard open-source libraries.

The README file should cover how to generate the environment needed to run the code, how to run the code, and any other necessary information.

The document should also cover the following aspects:

1. The approach used to generate the algorithm.
2. Proof of Correctness.
3. Complexity Analysis.

Evaluation Criteria:

1. The coverage and correctness of the algorithm across different circuit designs.
2. Space Complexity: extra space used by algorithm to generate the input vector needed.
3. Time Complexity: Time used by the algorithm to generate the input vector required.

Evaluation Rubrics:

- Code Quality (20%)
- Circuit Design (10%)
- Algorithm Design (50%) - Correctness and Time & Space Complexity
- Testing (20%)

Find Template to use below

(3 Pages Maximum from the template below)

2023 Girl Hackathon Ideathon Round: Solution Submission
Project Name:
Pariticipant Name:
Participant Email ID:
ReadMe File Links (Eg Github)
Brief summary Please summarize your problem statement and solution in a short paragraph.

Problem Statement

What are you doing, why, and for whom?

The approach used to generate the algorithm.

Proof of Correctness.

Complexity Analysis

Alternatives considered

Include alternate design ideas here which you are leaning away from.

References and appendices

Any supporting references, mocks, diagrams or demos that help portray your solution.

Any public datasets you use to predict or solve your problem.