VERIFICATION/VALIDATION

Javier Ramos Rodriguez

Assignment 1

#include <iostream.h>

// Assignment 1

//input: int a, int b
//output: int p=a^b

void main()
{
    int a,b,p=1;
    cout << "First number: "; cin >> a;
    cout << "Second number: "; cin >> b;

    while (b--) p*=a;
    cout << "Result: " << p;
}

- Easy to find fault: input b, cannot be 0.

- Hard to find fault: This program only works with small numbers (integer numbers), in c++ the compiler uses 32 bits to store the number and it can only store a value from -32768 to 32767.
Assignment 2

Failure sets:

- **Easy to find fault:**

![Easy to find fault graph](image1)

- **Hard to find fault:**

![Hard to find fault graph](image2)
Assigment 3

For each of your faults, calculate how many points are in the failure set and how many in the domain:

- **Easy to find fault:**

  **Failure set:** all points that satisfied \( \{ y < 0 \} \), been \( x, y \) the inputs.

  So, the failure set will have \( \frac{1}{2} \) of the domain points.

- **Hard to find fault:**

  **Failure set:** all points that satisfied \( \{ y < 0 \} \| \{ x^y > 32767 \} \), been \( x, y \) the inputs.

Determine a set of sub domains that are relevant for the program:

Because this program is so simple we can look just to two sub domains. When \( y < 0 \) and when \( y > 0 \).

In the first case, the program doesn’t work because the way it was programmed. So, the number of points in the failure set is equal to the number of points in the sub domain.

When \( y > 0 \), the program will work only with small numbers. So, the failure set will increase as \( x \) and \( y \) get bigger.