EXERCISE 1 – BRANCH PREDICTION (5 points)

In a standard 5-stage pipeline MIPS 32-bit architecture, with signed integer representation. Given the following assembly code, and a "bootstrapping" scenario where R0 is set to 1, R1 is set to 300.

```
LOOP:   LD      F3   0     (R0)
        ADDD    F1   F3    F3
        ADDI    R1   R1    3000
LOOP2:  MULTD   F2   F2    F3
        SUBI    R1   R1    3
        BNEZ    R1   LOOP2
        SUBI    R0   R0    2
        BNEZ    R0   LOOP
```

Answer the following questions:

Question 1: How many iterations for LOOP and LOOP2?

Answer 1:

The outer loop (LOOP) can be considered as an **infinite** loop because starting from 1, considering the SUBI of -2, it will never hit 0, hence we will treat this as the theoretical case of an asynthotic solution. On the other hand, the inner loop (LOOP2) is executed **1100 times** at the first iteration of LOOP, and **1001 times** at the following iterations of LOOP. However, being LOOP2 an inner loop of an infinite loop, also LOOP2 is executed will execute **infinite** times.

Question 2: Given a 1-bit BHT branch predictor, how many mispredictions are we going to observe?

Answer 2:

In case of 1-BHT, let's consider when the two branch instructions **collide (k=1).** We have two possible initial conditions: BHT bit equal to 0, i.e. Not Taken, or 1, Taken.

In the first case (NT), we have for LOOP2 an initial misprediction and a misprediction when we have to exit LOOP2. In the second case (T), we have a misprediction when we have to exit LOOP2. In both cases, the prediction bit at the exit from LOOP2 is NT, so the outer LOOP will always have a misprediction of Not Taken, hence a 100% failure rate.

When we are operatively, for the inner LOOP2 we have only one misprediction at the last iteration and still 100% of failure rate for the outer LOOP.

In case of 1-BHT, let's consider when the two branch instructions **do not collide (K>1).**

We have two possible initial conditions for the BHT bit used for LOOP2: equal to 0, i.e. Not Taken, or 1, Taken. In the first case (NT), we have for LOOP2 an initial misprediction and a misprediction when we have to exit LOOP2. In the second case (T), we have a misprediction when we have to exit LOOP2. When we are operatively, for the inner LOOP2, we have a misprediction at the entrance and a misprediction at the exit of LOOP2 (in line with the theoretical characteristics of 1-BHT).

We have two possible initial conditions for the BHT bit used for LOOP: equal to 0, i.e. Not Taken, or 1, Taken. In the first case (NT), we have for LOOP an initial misprediction. In the second case (T), we have a success. When we are operatively, for the outer LOOP, we have A 100% success rate.

Question 3: In changing the branch predictor, and in using a 2-bit BHT, how many mispredictions are we going to observe?

Answer 3:

Considering a 2-BHT , let's consider when the two branch instructions **collide (k=1) and to start from** an initial condition Strongly Taken (11). In this case, even if we have a misprediction at the exit of the inner LOOP2. In this way we go to Weak Taken (10) but we have an operatively success rate of 99.9% for the inner LOOP2 and 100% for the outer LOOP. Considering the other initial conditions, we get some initial mispredictions for LOOP2, but then we converge.

Question 4: Considering the answers to question 2 and 3, is the obtained result inline with theoretical characteristics of the two predictors?

Answer4:

Within this context we can see that this example is working as expected from a theoretical point of view, because the 1-bit BHT is converging to the worst case scenario, no matter the initialization while the 2-bit BHT is doing better.