Computer Architecture HW #3

Due: Wednesday, Sept. 22 (5 PM in ITT 305 mailbox or under my office door, ITT 313)

1. Consider the following insertion sort algorithm that sorts an array numbers:

```java
InsertionSort(numbers - address to integer array, length - integer)
integer firstUnsortedIndex, testIndex, elementToInsert;
for firstUnsortedIndex = 1 to (length - 1) do
    testIndex = firstUnsortedIndex - 1;
    elementToInsert = numbers[firstUnsortedIndex];
    while (testIndex >= 0) AND (numbers[testIndex] > elementToInsert) do
        numbers[testIndex + 1] = numbers[testIndex];
        testIndex = testIndex - 1;
    end while
    numbers[testIndex + 1] = elementToInsert;
end for
```

(a) (b) PREDICT_NOT_TAKEN

c) Assumptions:
- length = 100 and the numbers are initially in descending order before the insertion sort algorithm is called
- the five-stage pipeline of the text
- the outcome of conditional branches is known at the end of the ID stage
- target addresses of all branches is known at the end of the ID stage
- ignore any data hazards

Under the above assumptions, answer the following questions:

i) If fixed predict-never-taken is used by the hardware, then what will be the total branch penalty (# cycles wasted) for the algorithm? (Here assume NO branch target buffer)

```java
for cond testIndex > 0 while numbers[testIndex] > elementToInsert
4 4 x 99
= 10,496 cycles
end while
```

ii) If a branch target buffer with one history bit per entry is used, then what will be the total branch penalty (# cycles wasted) for the algorithm? (Assume predict-not taken is used if there is no match in the branch target buffer)

```
for cond testIndex > 0 while numbers[testIndex] > elementToInsert
4 4 + 8 x 98
= 796 cycles
```

iii) If a branch target buffer with two history bit per entry is used, then what will be the total branch penalty (# cycles wasted) for the algorithm? (Assume predict-not taken is used if there is no match in the branch target buffer)

```
for cond testIndex > 0 while numbers[testIndex] > elementToInsert
4 4 x 99
= 404 cycles
```