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

```plaintext
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
end InsertionSort
```

a) Where in the code would unconditional branches be used and where would conditional branches be used?

b) If the compiler could predict by opcode for the conditional branches (i.e., select whether to use machine language statements like: “BRANCH_LE_PREDICT_NOT_TAKEN” or “BRANCH_LE_PREDICT_TAKEN”), then which conditional branches would be "PREDICT_NOT_TAKEN" and which would be "PREDICT_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)

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)

   Explain your answer.

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)

   Explain your answer.