Lab Assignment 03

Task 1: Consider that you have given a problem of counting in a text T, the number of substrings that will start with a X and end with Y. For example text T= HXYX X DYBX in which there are four such substrings. Design more than one most efficient algorithm for the given problem and determine the efficiency of your proposed algorithms.

Task 2: Consider that a given set of n distinct numbers in which you have to find the length of the longest monotone increasing subsequence (i.e. subset of numbers which are strictly increasing from left to right). It does not require that the numbers be adjacent in the original set or that the longest sequence is unique.

For example, given a sequence

1 2 9 4 7 3 11 8 14 6

There are many monotone subsequences such as-

1 2 9
1 2
2 9
1 2 3 6
1 2 4 7 11 14

So longest monotone subsequence is

1 2 4 7 11 14

Design and implement an efficient algorithm to perform the above task as per the specification given. Generate program profile and analyze the running time of your algorithm.

Task 3: A poker deck contains 52 cards. Each card has a suit of either clubs, diamonds, hearts, or spades (denoted C, D, H, S in the input data). Each card also has a value of either 2 through 10, jack, queen, king, or ace (denoted 2, 3, 4, 5, 6, 7, 8, 9, T, J, Q, K, A). For scoring purposes card values are ordered as above, with 2 having the lowest and ace the highest value. The suit has no impact on value.

A poker hand consists of five cards dealt from the deck. Poker hands are ranked by the following partial order from lowest to highest.

High Card. Hands which do not fit any higher category are ranked by the value of their highest card. If the highest cards have the same value, the hands are ranked by the next highest, and so on.

Pair. Two of the five cards in the hand have the same value. Hands which both contain a pair are ranked by the value of the cards forming the pair. If these values are the same, the hands are ranked by the values of the cards not forming the pair, in decreasing order.

Two Pairs. The hand contains two different pairs. Hands which both contain two pairs are ranked by the value of their highest pair. Hands with the same highest pair are ranked by the value of their other pair. If these values are the same the hands are ranked by the value of the remaining card.

Three of a Kind. Three of the cards in the hand have the same value. Hands which both contain three of a kind are ranked by the value of the three cards.

Straight. Hand contains five cards with consecutive values. Hands which both contain a straight are ranked by their highest card.

Flush. Hand contains five cards of the same suit. Hands which are both flushes are ranked using the rules for High Card.
Full House. Three cards of the same value, with the remaining two cards forming a pair. Ranked by the value of the three cards.
Four of a Kind. Four cards with the same value. Ranked by the value of the four cards.
Straight Flush. Five cards of the same suit with consecutive values. Ranked by the highest card in the hand.
Your job is to compare several pairs of poker hands and to indicate which, if either, has a higher rank.

Input

The input file contains several lines, each containing the designation of ten cards: the first five cards are the hand for the player named "*Black*" and the next five cards are the hand for the player named "*White*".

Output

For each line of input, print a line containing one of the following:

```
Black wins.
White wins.
Tie.
```

Sample Input

```
2H 3D 5S 9C KD 2C 3H 4S 8C AH
2H 4S 4C 2D 4H 2S 8S AS QS 3S
2H 3D 5S 9C KD 2C 3H 4S 8C KH
2H 3D 5S 9C KD 2D 3H 5C 9S KH
```

Sample Output

```
White wins.
Black wins.
Black wins.
Tie.
```

Generate program profile and analyze the running time of your algorithm.