Beginning Level Problems

Problem 1: Cesar cipher

Cesar cipher uses a symmetric key to encrypt a message and decrypt a message. Suppose the key is 2, 'A' will be encrypted as 'C', 'B' will be encrypted as 'D', 'C' will be encrypted as 'E', …, and 'X' will be encrypted as 'Z', 'Y' will be encrypted as 'A' and 'Z' will be encrypted as 'B'.

Write a program that accepts input a text message that only contains capital English letters, punctuations, and spaces, and displays the decrypted message.

Note: (1) the key is constant and equal to 2; (2) punctuations and spaces will not be encrypted.

Sample input and output:

Input: K NQXG AQW
Output: I LOVE YOU

Input: EQFKPI KU HWP
Output: CODING IS FUN

Problem 2: Anagram checker

An anagram is a word or phrase formed by rearranging the letters of a different word or phrase, using all the original letters exactly once. Given two strings, check if they are anagrams of each other. Ignore the letter case.

Sample input and output:

Input:
String 1: REGAL
String 2: GLARE
Output: yes

Input:
String 1: BANNER
String 2: BARREN
Output: no

(Explanation: same letter set, but not the same repeated letters)

Solution hint: convert everything to lowercase, sort the two strings in alphabetical order, then compare them in order.

Problem 3: Accumulation of wealth

A family's wealth is accumulated from generation to generation. Suppose the following formula describes how fast a family could accumulate their wealth,

G(0) = 0; G(1) = 1; G(2) = 1; G(3) = 2; G(4) = 3, …, and G(n) = G(n-1) + G(n-2).

Write a program that accepts an input (generation number) and displays the accumulation of wealth of this generation.

Sample input and output:

Input: 1
Output: 1

Input: 3
Output: 2

Input: 8
Output: 21

Advanced Level Problems

Problem 1: Train swap

A train company organizes their trains in an order that is optimal for the delivery order in which the different cars will arrive at their respective destinations. A train is formed in a random order first, and then a crane is used to swap adjacent cars to achieve the optimal order,

Given a train where the cars are not in optimal order, and where you can only swap two adjacent cars at a time, find out the minimal number of swaps that are necessary to put the cars in order. The input is the initial order of the cars separated by a space and the output is the number of swaps.

Sample input and output:

Input: 1 3 2
Output: 1

Input: 4 3 2 1
Output: 6

Input: 3 1 2
Output: 2

Solution hint: the goal is to sort the numbers, so apply a bubble sort and count all the swaps that are made.

Problem 2: Rotate a matrix by 90 degrees
Given a matrix of size m*n (m rows and n columns), you task is to rotate it by 90 degrees in clockwise direction.

Sample input and output:

Input:
Number of rows: 3
Number of columns: 3
1 2 3
4 5 6
7 8 9

Output:
7 4 1
8 5 2
9 6 3

Input:
Number of rows: 4
Number of columns: 3
1 2 3
4 5 6
7 8 9
10 11 12

Output:
10 7 4 1
11 8 5 2
12 9 6 3

Problem 3: Match-4
A match-4 game consists of a table of 6 rows by 7 columns where players can insert tokens alternatively. A player wins the game if they have created a line of 4 consecutive tokens, either horizontally, vertically, or diagonally.

Given a configuration of a Match-4 game, determine if either of the opponents has won already. The input will be a 6x7 table containing the values 0 (empty), 1 (1st player), or 2 (2nd player). The output will have to be 0, if no player has won yet, 1 if the 1st player won, and 2 if the second one won. One can assume that only one player at most has won the game for any input board.

Sample input and output:

Input:
0 0 0 0 0 0 1
0 0 0 1 0 0 1
0 2 0 1 1 0 2
0 2 1 2 2 1 1
2 2 2 1 2 2 2
1 1 2 1 2 2 1
Output: 0

Input:
0 0 0 0 0 0 1
0 2 0 1 0 0 1
0 2 0 1 1 0 2
0 2 1 2 2 1 1
2 2 2 1 2 2 2
1 1 2 1 2 2 1
Output: 2