# C# Fundamentals – Sample (Easy) Intermediate Exam – March 2010

## Sample Problem 1 – Bitwise Operations

We are given an array of **n** 8-bit numbers (in range [0..255]), for example { 115, 8, 33 } and a natural number k, for example k=2. If the array of bytes is considered as a series of bits, the task is to rotate the bits with **k** positions rightwards. The rotation works as a simple bitwise shift, but the lost bits when shifting rightwards is transferred in the beginning of the sequence – on the left. The output is a sequence of **n** numbers – the resulted sequence from the right rotation with k positions.

**Note:** You are not allowed to convert the input into a sequence of bits (e.g. as string, as list or as array of bits or as other data structure holding the bits). You should process the input using bitwise operations.

|  |  |  |  |
| --- | --- | --- | --- |
| Sample input  |  | | --- | | **arr = { 115, 8, 33 }**  **k = 2** | | Sample output  |  | | --- | | **arr = { 92, 194, 8 }** | |

### Explanation

The input corresponds to the following sequence of bits:

|  |
| --- |
| **01110011 00001000 00100001** |

The output corresponds to the following sequence of bits:

|  |
| --- |
| **01011100 11000010 00001000** |

## Sample Problem 2 – Loops

We are given an array of n integer numbers. Write a program that prints all the sequences of three numbers, that the third number is equal to the sum of the first and the second. The array is cyclic i.e. after position n-1 is position 0. The sequences should be printed from left to right.

|  |  |  |  |
| --- | --- | --- | --- |
| Sample input  |  | | --- | | **arr = { 2, 2, 3, 5, 1, 1 }** | | Sample output  |  | | --- | | **2 + 3 = 5**  **1 + 1 = 2** | |

## Sample Problem 3 – Matrices

We are given a matrix of integer numbers with **n** rows and **m** columns, and a figure of the game Tetris. We can place the figure everywhere where we have only equal numbers under it. The figure is as follows:

Tetris-figure

The figure can be rotated to 90 degrees and can be inverted. Write a program that counts all the positions this figure can be placed on a matrix, so that the figure is over only equal numbers.

|  |  |  |  |
| --- | --- | --- | --- |
| Sample input  |  | | --- | | **matrix = {**  **{ 1, 2, 1, 1 },**  **{ 1, 1, 1, 0 },**  **{ 2, 1, 1, 1 },**  **{ 1, 2, 1, 5 },**  **}** | | Sample output  |  | | --- | | **6** | |

The 6 possible positions are as follows:

|  |  |  |
| --- | --- | --- |
| **{{ X, 2, 1, 1 },**  **{ X, X, 1, 0 },**  **{ 2, X, 1, 1 },**  **{ 1, 2, 1, 5 }}**  **{{ 1, 2, X, X },**  **{ 1, X, X, 0 },**  **{ 2, 1, 1, 1 },**  **{ 1, 2, 1, 5 }}** | **{{ 1, 2, X, 1 },**  **{ 1, X, X, 0 },**  **{ 2, X, 1, 1 },**  **{ 1, 2, 1, 5 }}**  **{{ 1, 2, 1, 1 },**  **{ X, X, 1, 0 },**  **{ 2, X, X, 1 },**  **{ 1, 2, 1, 5 }}** | **{{ 1, 2, 1, 1 },**  **{ 1, X, X, 0 },**  **{ 2, 1, X, X },**  **{ 1, 2, 1, 5 }}**  **{{ 1, 2, 1, 1 },**  **{ 1, X, 1, 0 },**  **{ 2, X, X, 1 },**  **{ 1, 2, X, 5 }}** |

## Sample Problem 4 – Simulation of the Game "War"

Write a program that simulates a simple “War” game with cards. The game is played by two players. Each of them initially has a set of cards. The players draw their top-of-the-deck card (the first player draws first). The player, that has the bigger card, gets the two cards and puts them in the bottom of the deck. The suit of the cards is of no matter, only the kind of the card.

If both players put equal cards a card war begins. Both players draw three cards. That is done like the following: the first player draws a card, then the second, then the first again etc. The card that matters is the third card. The player whose third card is bigger wins the war and gets all the eight cards (in the order of drawing) and puts them in their deck.

To be simpler if the third drawn cards are equal, if the turn is odd then the first player wins and if the turn is even – the second player wins. If a player should draw a card but he has no more cards – he loses. For example: if a player has two cards left and a war begins, he loses because he has not enough cards to play the war. If players don’t have enough card to play a war and such begins, then nobody wins (both players lose the game).

The winner is the player that wins all the cards and the other has no more cards. The program should print the winner and numbers of turns played.

Let assume that if a game has more than 3000 turns, the game stops and there is no winner.

The program input is given as string of cards, separated by intervals (only the kinds of cards, not the suits). The cards are not always a full deck (52 cards, 4 of each kind). The valid cards in increasing order are as follows: 2, 3, 4, 5, 6, 7, 8, 9, 0, J, Q, K, A ( to simplify we will use 0 as 10). This way each card is represented by a single symbol.

**Sample input (consists of two strings, each on separate line):**

|  |
| --- |
| **A 3 3 K Q J 4 4**  **2 4 3 8 2 6 2** |

That is the flow of a game:

After the first turn:

|  |
| --- |
| **3 3 K Q J 4 4 А 2**  **4 3 8 2 6 2** |

After the second turn:

|  |
| --- |
| **3 K Q J 4 4 А 2**  **3 8 2 6 2 3 4** |

On the third turn a war begins and the first player wins:

|  |
| --- |
| **4 4 А 2 3 3 K 8 Q 2 J 6**  **2 3 4** |

After the third turn:

|  |
| --- |
| **4 А 2 3 3 K 8 Q 2 J 6 4 2**  **3 4** |

After the fourth turn:

|  |
| --- |
| **А 2 3 3 K 8 Q 2 J 6 4 2 4 3**  **4** |

After the sixth turn the second player loses, because he has no more cards to play:

|  |
| --- |
| **2 3 3 K 8 Q 2 J 6 4 2 4 3 А 4**  **(no cards)** |

**Sample output:**

|  |
| --- |
| **6 turns, player 1 wins** |