Example Questions for Final Exam (Part 2)

1- Write a C# program that determines a given **sparse** matrix is an *anti-diagonal matrix* or not.

An ***anti-diagonal matrix*** is a matrix where all the entries are zero except those on the diagonal going from the lower left corner to the upper right corner (↗).

Example:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 5 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

```
int[,] s = new int[,] {{ 5, 5, 5 },{ 0, 4, 1 },{ 1, 3, 2 },{ 2, 2, 5 },{ 3, 1, 7 },{ 4, 0, -1 }};
```

2- Write a program that ***replaces all occurrences*** of a given word with a new one without using Replace function.

Example:

```
Input: "Bugün hava ÇoK soğuk ama çok soĞuk"
           "soğuk"   "sıcak"
Output: "bugün hava çok sıcak ama çok sıcak"
```

3- In a ***diving competition***, the final score is calculated by dropping the lowest and highest scores and then getting the remaining score.
Assume that there exists an array which stores the name of the competitors and their 3 judges' scores.

Write a program that prints the name of the winner, after discarding the high and low scores of two judges. A judge awards points between 1 (low) and 10 (high).

4- Write a recursive function to find:
How many ways can you choose *r* items from a set S of *n* distinct elements? $^{n}C_{r}$

$$^{n}C_{r} = {}^{n\text{-}1}C_{r} + {}^{n\text{-}1}C_{r\text{-}1} \quad | \quad n > r > 1$$
$$^{n}C_{n} = 1$$
$$^{n}C_{0} = 1$$

Example:

```
Input: n=4   r=2
Output:  6

S = {1, 2, 3, 4}
{1, 2}, {1, 3}, {1, 4}, {2, 3}, {2, 4}, {3, 4}
```

$^{n}C_{r}$ = n! / r! (n-r)!

Examples

$$\binom{13}{3} = 286 \qquad \binom{52}{5} = 2,598,960$$

5- Write a C# program that takes a number *n* from the user and constructs a string for ***factoring formulas***.

Factoring Formulas:

$$x^2 - y^2 = (x - y)(x + y)$$
$$x^3 - y^3 = (x - y)(x^2 + xy + y^2)$$
$$x^5 - y^5 = (x - y)(x^4 + x^3y + x^2y^2 + xy^3 + y^4\ )$$

6- One way to determine a long DNA sequence is to find the sequences of the pieces, and then *assemble* the pieces based on areas where they overlap with the same codes.

Write a **function** that produces a combined ***DNA string*** from two strings.

Example:
Input strings: aaagtgc   tgcgtg
Output string: aaagtgcgtg

Example:
Input strings: caaggctccta   taccc
Output string: caaggctcctaccc

Example:

```
Input strings:  tagctccta  tcctagg
Output string:  tagctcctagg
```

7- Write a program that determines an existing array has ***alternating parity*** or not.
The parity of an integer is 0 for even numbers and 1 for odd numbers.
An array has "*alternating parity*", if it alternates between *even* and *odd* numbers.

Example:

```
int[] a = {4, 3, 20, 9, 44, 65, 2, 1, 4}
Output: true
```

Example:

```
int[] a = {3, 14, 5, 2, 1, 10, 3, 8, 5, 13, 4, 3}
Output:  false
```

(because the array has two sequential odd numbers (5 and 13)).