Sukkur IBA University Khairpur Campus

Data Structures (Fall 2024)

Practice Exercises 01

Arrays

Task 01 (N consecutive Numbers)

Write a function that determine N consecutive same values. This function would return True if N consecutive values are same otherwise false.

Function name: boolean NConRep (int arr[][])

Example 1

Input: The array should be n*n. suppose n=4 then

2	1	5	10
33	33	33	33
4	7	16	43
11	12	22	40

Output: N is 4 so matrix is 4x4 and in 4 consecutive values are same so it should return True.

Example 2

Input: The array should be n*n. suppose n=3 then

12	2	45
12	34	2
12	1	23

Output: **N is 3** so matrix is 3x3 and in 3 consecutive values are same so it should return True.

Example 3

Input: The array should be n*n. suppose n=2 then

122	45
9	89

Output: **N is 2** so matrix is 2x2 and no any 2 consecutive values are same so it should return False.

Task 02 (Cryptography)

Write a functions that predict the unique key used to encrypt each character of a given message. Use an array to store the predicted keys. This keys can be used to decrypt the encrypted message back to the original message.

Function name: int[] predictKeys(String message, String encryptedMessage)

Example

String **message** = “Hello world”

The message above was encrypted in such a way that every character was encrypted after adding a unique key to it. The message after encryption become:

String **encryptedMessage** = “Igopt&^w{vo”

Ensure that your code handles the case where the lengths of message and encryptedMessage are not equal (in such cases, return an error message).

Hint: Use ASCII value manipulation to compute the keys and decrypt the message.

Task 03 (The Saddle Point)

A saddle point in a 2D array is an element that is the minimum in its row but the maximum in its column. Write a function to find a saddle point in a given 2D array, if it exists.

Write a function to find a saddle point in the array. If a saddle point exists, return its value and its position in the array.

Function name: findSaddlePoint(int[][] matrix)

Example 1

```
Input: int[][] matrix = {   {1, 2, 3},
                            {4, 5, 6},
                            {7, 8, 9}   };
```

Output: No saddle point

Example 2

```
Input: int[][] matrix = {   {3, 8, 4},
                            {9, 7, 6},
                            {2, 1, 5}   };
```

Output: Saddle Point: 7 at position (1, 1)

Example 3

```
Input: int[][] matrix = {   {3}   };
```

Output: Saddle Point: 3 at position (0, 0)

Task 04 (Matrix Transpose)

Write a function that transpose a given 2D array (matrix). Transposing a matrix involves flipping it over its diagonal, effectively swapping rows with columns.

Function name: int[][] transposeMatrix(int[][] matrix)

Example 1

Input:

```
int[][] matrix = {      {1, 2, 3},
                        {4, 5, 6}       };
```

Output: Transposed Matrix:

1 4

2 5

3 6

Example 2

```
Input: int[][] matrix = {       {7, 8},
                                {9, 10},
                                {11, 12}        };
```

Output: Transposed Matrix:

7 9 11

8 10 12