FIT1045 Algorithmic Problem Solving – Workshop 6.

Objectives

The **objectives of this workshop** are:

- To implement and manipulate data structures in Python.
- To implement algorithms on these structures in Python.
- To investigate the effect of the choice of representation of a solution on the algorithm.

Important: Complete the questions from last week's workshop if you have not already.

Task 0: Prelab

When performing arithmetic on matrices, we often want to take the product of the diagonal. Using the following example write a program that calculates the product of the main diagonal of a matrix and subtracts the reverse diagonal.

$$f\begin{pmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ c_0 & c_1 & c_2 \end{pmatrix} = a_0b_1c_2 - a_2b_1c_0$$

Task 1:

- Discuss some ways of representing a candidate solution to the N Queens problem.
- What are the advantages and disadvantages of each representation?
- In this workshop, we investigate two of the different representations. The first representation uses an $N \times N$ table where an entry is 1 if a Queen is placed at this position and 0 otherwise to represent placement of the Queens on the chessboard.

Write a Python program that takes as input from the user the dimension of the chessboard , N, and then the N positions of the Queens. The position of each Queen is given by two numbers (the row and column that the Queen is positioned in) separated by space where the rows/columns are numbered 0 to $N-1$. For example, if $N = 4$, the user input:
1 0
3 1
0 2
2 3
would place the queens at $T[1][0]$, $T[3][1]$, $T[0][2]$ and $T[2][3]$ in the table T. Your program must also check if the position entered by the user is not a valid position and ask the user to enter it again. For example, the position `4 3` is invalid if $N = 4$. Your program should store the candidate solution in a table T and print the table. (The table will be implemented as a list of lists in Python.)

For example: your program may do the following:

```
Enter N: 4
Enter position of Queen:   1 0
Enter position of Queen:   3 1
Enter position of Queen:   0 2
Enter position of Queen:   2 3
[[0, 0, 1, 0], [1, 0, 0, 0], [0, 0, 0, 1], [0, 1, 0, 0]]
```

Task 2:

Modify your program in Task 1 so that your program prints the table as a matrix. Your program should use a *function* called `printTable` that takes as input an $N \times N$ table T and prints the table as a matrix.

For example: your program would now do the following:

```
Enter N:  4
Enter position of Queen:   1 0
Enter position of Queen:   3 1
Enter position of Queen:   0 2
Enter position of Queen:   2 3
0  0  1  0
1  0  0  0
0  0  0  1
0  1  0  0
```

Task 3:

Modify your program in Task 1 so that it represents a candidate solution as a list L where $L[i]$ gives the row position of the Queen in column i. Print the list L.

For example: your program could do the following:

```
Enter N:  4
Enter position of Queen:   1 0
Enter position of Queen:   3 1
Enter position of Queen:   0 2
Enter position of Queen:   2 3
[1, 3, 0, 2]
```

Task 4:

Modify your program in Task 3, so that it prints the list L and then reprints it in matrix format. You must write a function `printListInTableFormat` that takes as input a list representing a candidate solution for the N-Queens problem and prints the solution in matrix format.

For example: your program could do the following:

```
Enter N:  4
Enter position of Queen:   1 0
Enter position of Queen:   3 1
Enter position of Queen:   0 2
Enter position of Queen:   2 3
[1, 3, 0, 2]
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0
```

Task 5:

Modify your program in Task 4, so that it prints a message telling the user if their candidate solution is a correct solution to the N-Queens problem. You should write a function `checkSolution` that takes the list representing the candidate solution and returns True if the candidate solution is a correct solution and False otherwise. You may like to write additional functions that can be used by the function `checkSolution`.

For example: your program could do the following:

```
Enter N:  4
Enter position of Queen:   1 0
Enter position of Queen:   3 1
Enter position of Queen:   0 2
Enter position of Queen:   2 3
[1, 3, 0, 2]
0 0 1 0
```

```
1 0 0 0
0 0 0 1
0 1 0 0
This is a real solution to the problem.
```

and another example . . .

```
Enter N:  3
Enter position of Queen:  0 1
Enter position of Queen:  1 2
Enter position of Queen:  2 0
[2, 0, 1]
0 1 0
0 0 1
1 0 0
This is not a real solution to the problem.
```

Which representation was easier to print as a matrix?
Which representation would make it easier to check if the candidate solution was a correct solution to the problem?