

# Connect DNA

For this task, we have devised an extension of the popular game Four-in-a-Row. Like the original game, our expansion is also played on a rectangular  $m \times n$  board game with  $m$  rows and  $n$  columns. In contrast to the original Four-in-a-Row game, our expansion can be played with more than two players. Each player must choose a letter of the alphabet to denote his pieces.

In the example below, we illustrate the extension of the Four-in-a-Row game with a fully stocked  $9 \times 8$  game board on which four players have respectively chosen the letters A, C, G and T as their playing pieces. That's why we named this variant DNA-in-a-row.

## Assignment

### 1. Write a function

```
findSolutions(gameboard)
```

that indicates the position of a Four-in-a-Row on a given board. The board must be passed to this function as a list of lists. Here, each element of the outer list is a row of the board. A row of the board is represented as a list of capital letters.

The function must return a tuple with all Four-in-a-Row positions. Each element of this tuple is a Four-in-a-Row position which is also presented as a tuple itself. Each set of four horizontal consecutive identical characters is represented as a tuple, the first two elements indicating the row and column number of the leftmost position of the series of four consecutive identical characters. The third element of the tuple is composed of the string horizontally. Analogously, each set of four vertically successive identical characters is represented as a tuple of three elements, of which the third element contains the string vertically, and the first two elements correspond with the row and column number of the upper position in the series of four successive equal letters. The elements of the tuple that is returned as the result of the function must be sorted in ascending order.

### 2. Write a function

```
showSolutions(gameboard)
```

that prints both a formatted representation of the board as all positions with Four-in-a-Row to the output. To this function, a game board must be passed as argument, which is the same size as described in the `findSolutions` function.

The printing of the board should be in the format illustrated in the example below. On the first row of the column numbers are printed above the columns of the board. The following rows each start with a row number followed by the letters that are printed to the corresponding positions of the board. Row and column numbers are indexed from 1. For each column of the output five character positions are reserved to be filled right-aligned.

After showing the game board the positions with four consecutive identical characters are also printed. Obviously the function `showSolutions` will have to make use of the `findSolutions` function. The positions should be printed in the same order like in the tuple that is returned by the `findSolutions` function. The row and column number of each solution is written on a separate line, followed by a space and the direction of the solution (horizontally or vertically).

## Example

```
>>> game board = [['T', 'T', 'T', 'A', 'C', 'C', 'G', 'A'],
...               ['C', 'C', 'G', 'T', 'C', 'G', 'T', 'A'],
...               ['T', 'C', 'A', 'T', 'T', 'G', 'T', 'G'],
...               ['T', 'G', 'C', 'G', 'C', 'G', 'C', 'G'],
...               ['C', 'G', 'T', 'G', 'C', 'C', 'G', 'T'],
...               ['A', 'T', 'T', 'T', 'T', 'G', 'G', 'C'],
...               ['A', 'T', 'C', 'T', 'T', 'A', 'T', 'A'],
...               ['A', 'T', 'G', 'G', 'C', 'T', 'T', 'C'],
...               ['T', 'T', 'C', 'A', 'C', 'C', 'G', 'T']]
```

```
>>> findSolutions(game board)
((6, 2, 'horizontally'), (6, 2, 'vertically'))
```

```
>>> showSolutions(game board)
```

	1	2	3	4	5	6	7	8
1	T	T	T	A	C	C	G	A
2	C	C	G	T	C	G	T	A
3	T	C	A	T	T	G	T	G
4	T	G	C	G	C	G	C	G
5	C	G	T	G	C	C	G	T
6	A	T	T	T	T	G	G	C
7	A	T	C	T	T	A	T	A
8	A	T	G	G	C	T	T	C
9	T	T	C	A	C	C	G	T

```
6,2 horizontally
```

```
6,2 vertically
```

Voor deze opgave hebben we een uitbreiding bedacht op het bekende vier-op-een-rij spel. Net als het originele spel, wordt onze uitbreiding ook op een rechthoekig  $m \times n$  spelbord gespeeld met  $m$  rijen en  $n$  kolommen. In tegenstelling tot het originele vier-op-een-rij spel kan onze uitbreiding echter met meer dan twee spelers gespeeld worden. Elke speler kiest hierbij een letter uit het alfabet om zijn speelstukken aan te duiden.

In onderstaand voorbeeld illustreren we deze uitbreiding op het vier-op-een-rij spel met een volledig gevuld  $9 \times 8$  spelbord, waarop vier spelers respectievelijk de letters A, C, G en T gekozen hebben om hun speelstukken aan te duiden. Vandaar dat we deze variant DNA-op-een-rij gedoopt hebben.

## Opgave

### 1. Schrijf een functie

```
zoekOplossingen(spelbord)
```

die aangeeft op welke posities van een gegeven spelbord vier-op-een-rij wordt aangetroffen. Het spelbord moet aan deze functie doorgegeven worden als een lijst van lijsten. Hierbij stelt elk element van de buitenste lijst een rij van het spelbord voor. Een rij van het spelbord wordt voorgesteld als een lijst van hoofdletters.

De functie moet een tuple met alle vier-op-een-rij posities teruggeven. Elk element van deze tuple is een vier-op-een-rij positie die zelf ook als een tuple wordt voorgesteld. Elke reeks van vier horizontaal opeenvolgende gelijke letters wordt voorgesteld als een tuple, waarvan de eerste twee elementen het rij- en kolomnummer aangeven van de meest linkse positie van de reeks van vier opeenvolgende gelijke letters. Het derde element van de tuple bestaat uit de string `horizontaal`. Analoog wordt elke reeks van vier verticaal

opeenvolgende gelijke letters voorgesteld als een tuple van drie elementen, waarvan het derde element de string verticaal bevat, en de eerste twee elementen overeenkomen met het rij- en kolomnummer van de bovenste positie uit de reeks van vier opeenvolgende gelijke letters. De elementen van het tuple dat als resultaat van de functie wordt teruggegeven, moeten in oplopende volgorde gesorteerd worden.

## 2. Schrijf een functie

toonOplossingen(spelbord)

die zowel een opgemaakte voorstelling van het spelbord als alle posities waarop vier-op-een-rij wordt aangetroffen naar de uitvoer schrijft. Aan deze functie moet een spelbord als argument doorgegeven worden, dat hetzelfde formaat heeft als beschreven bij de functie zoekOplossingen.

Het uitschrijven van het spelbord moet gebeuren in het formaat dat geïllustreerd wordt in onderstaande voorbeeld. Op de eerste rij worden de kolomnummers uitgeschreven boven de kolommen van het spelbord. De volgende rijen starten telkens met een rijnummer gevolgd door de letters die ingevuld zijn op de corresponderende posities van het spelbord. Rij- en kolomnummers worden hierbij geïndexeerd vanaf 1. Voor elke kolom van de uitvoer worden vijf karakterposities gereserveerd, die rechts uitgelijnd moeten ingevuld worden.

Na de weergave van het spelbord worden ook de posities uitgeschreven waarop vier opeenvolgende gelijke letters aangetroffen worden. Uiteraard moet de functie toonOplossingen hiervoor gebruik maken van de functie zoekOplossingen. De posities moeten in dezelfde volgorde worden uitgeschreven als deze in het tuple dat door de functie zoekOplossingen wordt teruggegeven. Het rij- en kolomnummer van elke oplossing wordt uitgeschreven op een afzonderlijke regel, gevolgd door een spatie en de richting van de oplossing (horizontaal of verticaal).

## Voorbeeld

```
>>> spelbord = [['T', 'T', 'T', 'A', 'C', 'C', 'G', 'A'],
...             ['C', 'C', 'G', 'T', 'C', 'G', 'T', 'A'],
...             ['T', 'C', 'A', 'T', 'T', 'G', 'T', 'G'],
...             ['T', 'G', 'C', 'G', 'C', 'G', 'C', 'G'],
...             ['C', 'G', 'T', 'G', 'C', 'C', 'G', 'T'],
...             ['A', 'T', 'T', 'T', 'T', 'G', 'G', 'C'],
...             ['A', 'T', 'C', 'T', 'T', 'A', 'T', 'A'],
...             ['A', 'T', 'G', 'G', 'C', 'T', 'T', 'C'],
...             ['T', 'T', 'C', 'A', 'C', 'C', 'G', 'T']]
```

```
>>> zoekOplossingen(spelbord)
((6, 2, 'horizontaal'), (6, 2, 'verticaal'))
```

```
>>> toonOplossingen(spelbord)
  1  2  3  4  5  6  7  8
1  T  T  T  A  C  C  G  A
2  C  C  G  T  C  G  T  A
3  T  C  A  T  T  G  T  G
4  T  G  C  G  C  G  C  G
5  C  G  T  G  C  C  G  T
6  A  T  T  T  T  G  G  C
7  A  T  C  T  T  A  T  A
8  A  T  G  G  C  T  T  C
9  T  T  C  A  C  C  G  T
```

6,2 horizontaal

6,2 verticaal

