

15 puzzle

A 15 puzzle consists of a rectangular board with m rows and n columns, where one of the fields stays empty. Puzzle pieces on horizontally or vertically adjacent fields can be moved to this empty field. Because of this, the whole field moves. The first 15 puzzle was invented by Noyes Chapman and started a real puzzle mania in 1880.



The 15 puzzle invented by Noyes Chapman consisted of a 4×4 grid of which the field are numbered from 1 to 15.

In various variants of the puzzle, the pieces are marked with numbers, letters or parts of an image. The purpose is to first move the pieces a large number of times, so that they are positioned in a random order on the board. After that, the pieces must be moved in a way that they end up in the original order on the board.

Assignment

Define a class `fifteenpuzzle` to represent 15 puzzles, of which the pieces are marked with ASCII characters. This class must implement the following methods:

- An initializing method `__init__` to which three arguments must be given: *i*) the number of rows m , *ii*) the number of columns n and *iii*) a string. The pieces of the puzzle must be marked from left to right and from top to bottom with the consecutive characters of the given string. This string must have an equal amount of characters as the number of fields in the 15 puzzle, and must have exactly one space that indicates the position of the empty field. The initializing method must raise an `AssertionError` with the text `invalid configuration` if the condition is not met. The board must also have at least three rows and three columns, if not, the initializing method must raise an `AssertionError` with the text `puzzle must contain at least three rows and columns`.
- A method `__str__` with which a string representation of the current state of the 15 puzzle can be printed. The string representation consists of m lines, which n characters are written out that mark the puzzle pieces. The string representation itself does not end in a newline.
- A method `__repr__` with which a string representation of the current state of the 15 puzzle can be printed. This string representation reads as a Python expression that makes a new object of the class `fifteenpuzzle` that has the same condition.
- A method `move` with which the empty field of the 15 puzzle is moved successively over a number of given positions. The possible directions are left (L), right (R), up (U) and down (D) and indicate in which direction the empty field must be moved, not the direction in which an adjacent puzzle piece is moved towards the empty field. To this method, a string must be given that solely consists of the L, R, U and D. Based on these directions, the method must execute the corresponding movement of the empty field. As soon as an invalid movement must be executed, the method must raise an `AssertionError` with the text `invalid direction`. Previous (correct) movements, however, must still be executed.

Example

The example below shows how a 4×4 15 puzzle of which the letters initially read pikante loketten, can be converted over a number of movements to a puzzle of which the letters spell the word patentektoniek.

```
>>> puzzle = Slidingpuzzle(4, 4, 'o draconiandevil'); print(puzzle)
o dr
acon
iand
evil
>>> puzzle.slide('L'); print(puzzle)
odr
acon
iand
evil
>>> puzzle.slide('RRR'); print(puzzle)
odr
acon
iand
evil
>>> puzzle.slide('R')
Traceback (most recent call last):
AssertionError: invalid direction: R
>>> puzzle.slide('LLDRURULLDRLDRRLURLDRULRRUDDLRLURDU'); print(puzzle)
leon
ardo
davi
nci
>>> puzzle
Slidingpuzzle(4, 4, 'leonardodavinci ')

>>> puzzle = Slidingpuzzle(2, 2, "AOJN")
Traceback (most recent call last):
AssertionError: puzzle must have at least 3 rows and columns
>>> puzzle = Slidingpuzzle(4, 4, "AOJNDH")
Traceback (most recent call last):
AssertionError: invalid configuration
>>> puzzle = Slidingpuzzle(4, 4, "OEKBFJOZKSOLDFKE")
Traceback (most recent call last):
AssertionError: invalid configuration
```

Below you will find a graphical display of this 15 puzzle.



Voorbeeld van een schuifpuzzel waarvan de puzzelstukken gemarkeerd zijn met letters. Als je van links naar rechts, en van boven naar onder de letters achter elkaar zet, dan wordt over een aantal schuifbeurten de tekst pikante loketten omgezet in het woord patentektoniek.

Omdat we het niet kunnen laten, staan hieronder nog enkele voorbeelden van schuifpuzzels die in een aantal schuifbeurten wartaal (of toch niet?) omzetten in een herkenbaar woord.

```
>>> puzzle = Slidingpuzzle(3, 6, 'risereems npatient')
```

```
>>> puzzle.slide('DUDURRRDLRDLRUDUDULLRDRLR')
>>> print(puzzle)
misrep
resent
ation

>>> puzzle = Slidingpuzzle(7, 3, 'pigheneo iolnrpilceps')
>>> puzzle.slide('DLRDLLUDUUURRLDUDDRDLRDDULLRRLDR')
>>> print(puzzle)
pig
eon
hol
epr
inc
ipl
es

>>> puzzle = Slidingpuzzle(4, 4, 'penitential moms')
>>> puzzle.slide('LLLURDDLRRUULLDRDRUURDLUURDLLURDLLURDDDRR')
>>> puzzle
Slidingpuzzle(4, 4, 'implementations ')

>>> puzzle = Slidingpuzzle(4, 4, 'mythical gorilla')
>>> puzzle.slide('DRRUURDDLURDLLURULDRURDDLURDDLUULLURRRDDLLLUURRRDD')
>>> puzzle
Slidingpuzzle(4, 4, 'algorithmically ')

```

Een schuifpuzzel bestaat uit een rechthoekig bord van m rijen en n kolommen, waarbij één van de velden leeg blijft. Puzzelstukken op horizontaal of verticaal aangrenzende velden kunnen naar dit lege veld geschoven worden. Hierdoor verschuift in essentie het lege veld, vandaar de naam van de puzzel. De eerste schuifpuzzel werd uitgevonden door Noyes Chapman en ontketende in 1880 een echte puzzelgekte.



De schuifpuzzel uitgevonden door Noyes Chapman bestond uit een 4×4 bord waarvan de velden genummerd waren van 1 tot en met 15.

Bij verschillende varianten van de schuifpuzzel worden puzzelstukken gemarkeerd met getallen, letters of stukjes van een afbeelding. Het spel bestaat er dan in om de puzzelstukken eerst een groot aantal keer te verschuiven, zodat ze in een schijnbaar willekeurige volgorde op het bord staan, en daarna te proberen om de stukken terug in de oorspronkelijke volgorde op het bord te schuiven.

Opgave

Definieer een klasse `Schuifpuzzel` waarmee schuifpuzzels kunnen voorgesteld worden, waarvan de puzzelstukken gemarkeerd zijn met ASCII karakters. Deze klasse moet volgende methoden implementeren:

- Een initialisatiemethode `__init__` waaraan drie argumenten moeten meegegeven worden: *i*) het aantal rijen m , *ii*) het aantal kolommen n en *iii*) een string. De puzzelstukken van de

schuifpuzzel moeten van links naar rechts, en van boven naar onder gemarkeerd worden met de opeenvolgende karakters van de gegeven string. Deze string moet dus juist evenveel karakters tellen als het aantal velden van de schuifpuzzel, en er moet juist één spatie in voorkomen die de positie van het lege veld aangeeft. De initialisatiemethode moet een `AssertionError` met de tekst `ongeldige configuratie` opwerpen indien niet aan deze voorwaarden voldaan is. Het spelbord moet ook minstens drie rijen en kolommen tellen, en de initialisatiemethode moet een `AssertionError` met de tekst `puzzel moet minstens drie rijen en kolommen hebben` opwerpen als dat niet het geval is.

- Een methode `__str__` waarmee een stringvoorstelling van de huidige toestand van de schuifpuzzel kan uitgeschreven worden. Deze stringvoorstelling bestaat uit `m` regels, waarbij op elke regel de `n` karakters die de puzzelstukken markeren achter elkaar uitgeschreven worden. Deze stringvoorstelling eindigt zelf niet op een newline.
- Een methode `__repr__` waarmee een stringvoorstelling van de huidige toestand van de schuifpuzzel kan uitgeschreven worden. Deze stringvoorstelling leest als een Python expressie die een nieuw object aanmaakt van de klasse `Schuifpuzzel` dat dezelfde toestand heeft.
- Een methode `schuif` waarmee het lege veld van de schuifpuzzel achtereenvolgens over een aantal gegeven richtingen verschoven wordt. De mogelijke richtingen zijn links (L), rechts (R), op (O) en neer (N) en geven aan in welke richting het lege veld verschoven wordt, en dus niet de richting waarin een aangrenzend puzzelstuk naar het lege veld geschoven wordt. Aan deze methode moet een string doorgegeven worden, die enkel bestaat uit de letters L, R, O en N. Op basis van deze richtingen moet de methode achtereenvolgens de corresponderende schuifbewegingen van het lege veld doorvoeren. Van zodra een ongeldige schuifbeweging moet uitgevoerd worden, moet de methode een `AssertionError` met de tekst `ongeldige richting` opwerpen. Voorgaande (geldige) schuifbewegingen moeten echter wel uitgevoerd worden.

Voorbeeld

Onderstaand voorbeeld toont hoe je een `4 \times 4` schuifpuzzel waarvan de letters initieel lezen als pikante loketten, over een aantal schuifbeurten kunt omvormen tot een puzzel waarvan de letters het woord platentektoniek spellen.

```
>>> puzzel = Schuifpuzzel(4, 4, 'pikante loketten'); print(puzzel)
pika
nte
loke
tten
>>> puzzel.schuif('L'); print(puzzel)
pika
nt e
loke
tten
>>> puzzel.schuif('LL'); print(puzzel)
pika
nte
loke
tten
>>> puzzel.schuif('L')
Traceback (most recent call last):
AssertionError: ongeldige richting: L
>>> puzzel.schuif('ORRRNLLLORNNLLOORRRNLLNNLORNROOOLNRRNLLLORRNRN'); print(puzzel)
plat
ente
kton
```

```
iek
>>> puzzel
Schuifpuzzel(4, 4, 'platentektoniek ')
```

```
>>> puzzel = Schuifpuzzel(2, 2, "AOJN")
Traceback (most recent call last):
AssertionError: puzzel moet minstens 3 rijen en kolommen hebben
>>> puzzel = Schuifpuzzel(4, 4, "AOJNDH")
Traceback (most recent call last):
AssertionError: ongeldige configuratie
>>> puzzel = Schuifpuzzel(4, 4, "OEKBFJOZKSOLDFKE")
Traceback (most recent call last):
AssertionError: ongeldige configuratie
```

Hieronder zie je de grafische voorstelling van deze schuifpuzzel.



Voorbeeld van een schuifpuzzel waarvan de puzzelstukken gemarkeerd zijn met letters. Als je van links naar rechts, en van boven naar onder de letters achter elkaar zet, dan wordt over een aantal schuifbeurten de tekst pikante loketten omgezet in het woord platentektoniek.

Omdat we het niet kunnen laten, staan hieronder nog enkele voorbeelden van schuifpuzzels die in een aantal schuifbeurten wartaal (of toch niet?) omzetten in een herkenbaar woord.

```
>>> puzzel = Schuifpuzzel(3, 6, 'tardwean schpeaper')
>>> puzzel.schuif('LRLNLRROLNLORLNOORNRLRNRLRRONRR')
>>> print(puzzel)
aardwe
tensch
apper
```

```
>>> puzzel = Schuifpuzzel(7, 3, 'catal uteitsiripcinpe')
>>> puzzel.schuif('LONOLNNONNNOONNRLNRRLRLR')
>>> print(puzzel)
act
ual
ite
its
pri
nci
pe
```

```
>>> puzzel = Schuifpuzzel(4, 4, 'migraine toestel')
>>> puzzel.schuif('RROOLNNROORNNLNROLLLNRRROLLORRRNLNLLNROOORNRLNLR')
>>> puzzel
Schuifpuzzel(4, 4, 'meteorietinslag ')
```

```
>>> puzzel = Schuifpuzzel(4, 4, 'eenstemmig opaal')
>>> puzzel.schuif('LLORORNRRNNLOOROLNRRNNLLORNLOLORRNLLNROOLNROOLNRRROLNRRROLNRRNNLLLOORRRNN')
>>> puzzel
Schuifpuzzel(4, 4, 'paleomagnetisme ')
```