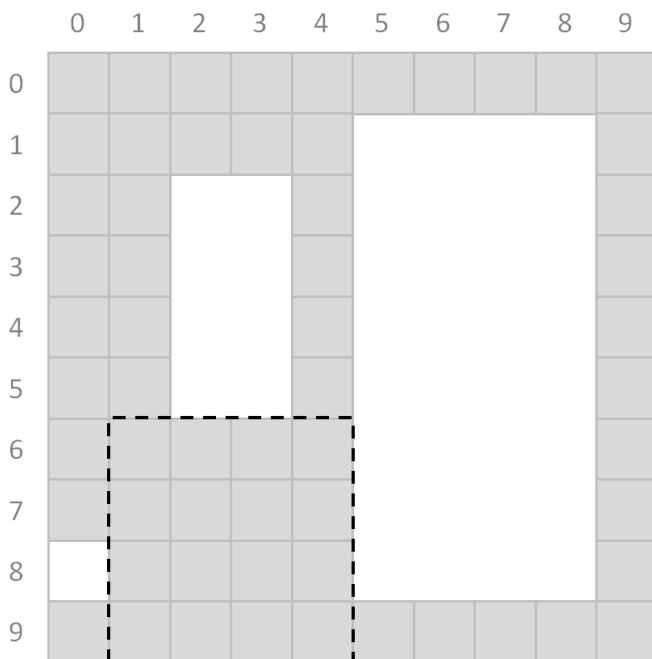


Allotment

In an *allotment*, a piece of land is divided in various *lots* that each have a different owner. Farmers used to have small pieces of land, that were often intertwined. Back then, a farmer had to walk or drive through the land of another farmer in order to reach his own lots. Today, because of the expansion in agriculture, square lots are used, and farmers strive to work with *parcels* that are as large as possible: rectangular areas of adjacent lots.

The example below shows a rectangular piece of land that was divided in 10×10 square lots. Three parcels (white triangles) are already taken into use by farmers. Search the surface for the largest possible parcel of which no lots are being used. In this example, this is a parcel that consists of 16 lots, as indicated by the striped rectangle.



Assignment

Define a class `Allotment` with the following methods:

- An initializing method `__init__` with which an allotment can be made out of a rectangular piece of land. This allotment consists of a $m \times n$ grid of lots, with m the number of rows and n the number of columns in the grid. The values m and n must be given to the initializing method as arguments. Initially, there are no reserved lots (appointed to an owner).
- A method `__str__` with which the current reservations of the lots from the allotment can be given. No arguments can be given to this method. A reserved lot is indicated with a hyphen (-). A lot that has not yet been reserved, is indicated with a hash (#). Below you will find some examples of how the entire allotment should be printed.
- A method `reserve` with which a rectangular parcel can be reserved. This method has four parameters, to which integers can be given. The first two parameters represent the row number and the column number of the position of the left upper corner of the parcel, and are obligatory. The last two parameters are optional, and represent the row number and the column number of the position of the right lower corner of the parcel. If these last

parameters are not given, the parcel consists of only one lot, of which the position is indicated by the first two parameters. As indicated in the image above, the rows are numbered from top to bottom and the columns from left to right, each starting from zero. If the parcel contains lots that have already been reserved, no lots must be reserved, and the method must raise an `AssertionError` as illustrated in the example below.

- A method `largestParcel` that prints the size of the largest parcel that can still be reserved. I.e. this is the size of the largest rectangular area that consists of lots that have not yet been reserved. No arguments can be given to this method.

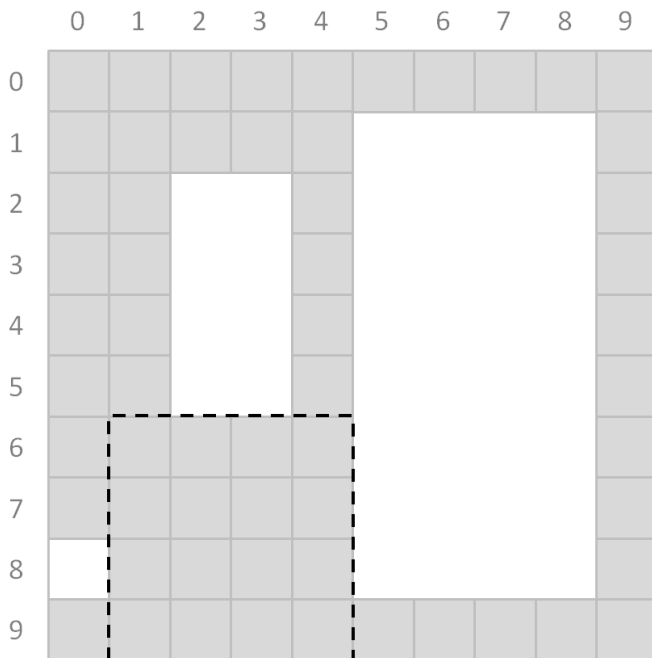
Example

```
>>> allotment = Allotment(6, 6)
>>> print(allotment)
#####
#####
#####
#####
#####
#####
>>> allotment.largestParcel()
36
>>> allotment.reserve(3, 0, 5, 2)
>>> print(allotment)
#####
#####
#####
---###
---###
---###
>>> allotment.largestParcel()
18
>>> allotment.reserve(0, 3, 2, 5)
>>> print(allotment)
###--
###--
###--
---###
---###
---###
>>> allotment.largestParcel()
9
>>> allotment.reserve(2, 2, 3, 3)
Traceback (most recent call last):
AssertionError: parcel can not be reserved
>>> print(allotment)
###--
###--
###--
---###
---###
---###
```

Bij een *verkaveling* wordt een stuk land opgedeeld in verschillende *kavels* die elk een andere eigenaar kunnen hebben. Vroeger hadden de boeren kleine stukken land, die vaak door elkaar heen lagen. Een boer moest toen over het land van een andere boer lopen of rijden om naar zijn eigen kavels te gaan. Tengevolge van de schaalvergroting van de landbouw wordt tegenwoordig

gewerkt met vierkante kavels, en streven boeren ernaar om te werken met zo groot mogelijke *percelen*: rechthoekige gebieden van aaneengesloten kavels.

Onderstaand voorbeeld toont een rechthoekig stuk land dat werd opgedeeld in 10×10 vierkante kavels. Drie percelen (witte rechthoeken) werden reeds door boeren in gebruik genomen. Zoek de oppervlakte van het grootst mogelijke perceel waarvan nog geen enkel kavel in gebruik werd genomen. In dit voorbeeld is dit een perceel dat bestaat uit 16 kavels, zoals aangegeven door de gestreepte rechthoek.



Opgave

Definieer een klasse `Verkaveling` met volgende methoden:

- Een initialisatiemethode `__init__` waarmee een verkaveling kan aangemaakt worden van een rechthoekig stuk land. Deze verkaveling bestaat uit een $m \times n$ rooster van kavels, met m het aantal rijen en n het aantal kolommen in het rooster. De waarden m en n moeten als argumenten aan de initialisatiemethode doorgegeven worden. Initieel wordt nog geen enkel kavel gereserveerd (toegewezen aan een eigenaar).
- Een methode `__str__` waarmee de huidige reservaties van de kavels uit de verkaveling kunnen weergegeven worden. Aan deze methode kunnen geen argumenten meegegeven worden. Een gereserveerd kavel wordt aangegeven met een koppelteken (-). Een kavel dat nog niet gereserveerd werd, wordt aangegeven met een hekje (#). Hieronder vind je enkele voorbeelden van hoe de volledige verkaveling moet weergegeven worden.
- Een methode `reserveer` waarmee een rechthoekig perceel kan gereserveerd worden. Deze methode heeft vier parameters, waaraan natuurlijke getallen kunnen doorgegeven worden. De eerste twee parameters stellen het rij- en kolomnummer van de positie van de linkerbovenhoek van het perceel voor, en moeten verplicht doorgegeven worden. De laatste twee parameters zijn optioneel, en stellen het rij- en kolomnummer van de positie van de rechteronderhoek van het perceel voor. Indien deze laatste parameters niet worden doorgegeven, dan bestaat het perceel uit slechts één kavel, waarvan de positie wordt aangegeven door de eerste twee parameters. Zoals aangegeven in bovenstaande afbeelding worden de rijen genummerd van boven naar onder en de kolommen van links

naar rechts, waarbij de nummering steeds start vanaf nul. Indien het perceel kavels bevat die reeds gereserveerd werden, dan moet er geen enkel kavel gereserveerd worden, en moet de methode een AssertionError opwerpen zoals geïllustreerd in onderstaand voorbeeld.

- Een methode `grootstePerceel` die de grootte teruggeeft van het grootste perceel dat nog kan gereserveerd worden. Dit is dus de grootte van het grootste rechthoekige gebied dat bestaat uit kavels die nog niet gereserveerd werden. Aan deze methode kunnen geen argumenten doorgegeven worden.

Voorbeeld

```
>>> verkaveling = Verkaveling(6, 6)
>>> print(verkaveling)
#####
#####
#####
#####
#####
#####
>>> verkaveling.grootstePerceel()
36
>>> verkaveling.reserveer(3, 0, 5, 2)
>>> print(verkaveling)
#####
#####
#####
---###
---###
---###
>>> verkaveling.grootstePerceel()
18
>>> verkaveling.reserveer(0, 3, 2, 5)
>>> print(verkaveling)
###--
###--
###--
---###
---###
---###
>>> verkaveling.grootstePerceel()
9
>>> verkaveling.reserveer(2, 2, 3, 3)
Traceback (most recent call last):
AssertionError: perceel kan niet gereserveerd worden
>>> print(verkaveling)
###--
###--
###--
---###
---###
---###
```