

Avantari Technologies Private Limited

AI Developer Test

Objective

The main objective of this test is to gauge the following:

- Logical Reasoning
- Problem Solving
- Ability to Debug your code
- Python Skills

To assess these qualities we require you to solve the following test in Python. You may submit your answers to deljohn@avantari.org and bhairav@avantari.org with the subject "AI Developer Test".

Time and Submission Limit

You have exactly **48 hours** to answer the chosen test. After which we will require you to submit your results by sending an email to the above email addresses.

Once you submit the test, if you're unsuccessful the first time, we will give you a second attempt and also inform you where you have not met the desired objectives. You will have 24 hours to be successful in your second attempt.

There will be no more chances beyond the second attempt.

At anytime during the tests if you have a question, you may send an email to deljohn@avantari.org and we will reply to you within 24 hours. Ask your questions early, and do not submit unfinished code.

Task

AIM

This test measures your logical skills and your ability to visualise a problem. We are also looking at your python writing skills, your ability to structure code and since the problem is complex, we want you to debug your own code.

Problem Statement

There is a partially constructed platform floating in space which is made of cubes. The platform can be represented by 2D array with each element representing number of cubes in respective position.

The way the platform is piled up, is given by an $n \times m$ matrix, where each element says how many cubes are stacked upward in that position. You have to write a function for calculating the quantity of water which will be stored in that construction if we pour unlimited supply of water on top of it. Since the platform is suspended in space, if an element has **0** as one of the elements, it will represent a drain. Use Numpy to solve the problem.

```
def WaterStoredInPlatform(platform):
```

Deliverables

- 1) where the function `WaterStoredInPlatform()` is defined.
- 2) A test file which will call `WaterStoredInPlatform()` with multiple test cases

Examples:

To elaborate on the problem, and to indicate how to solve the question, we have indicated the expected output from a given input matrix. **This does not mean that your code should work only for these case.** It should work for all cases, as **we will be checking your logic not your results.**

Example I:

```
[2 . 2 . 2]  
[2 . 2 . 2]  
[2 . 2 . 2]
```

Indicates a cuboid of 3x3 area units, which cannot retain any amount of water (0 units of water retained).

Example II:

A 4x3 matrix like the following

```
[2 . 2 . 2 . 2]  
[2 . 1 . 2 . 1]  
[2 . 2 . 2 . 1]
```

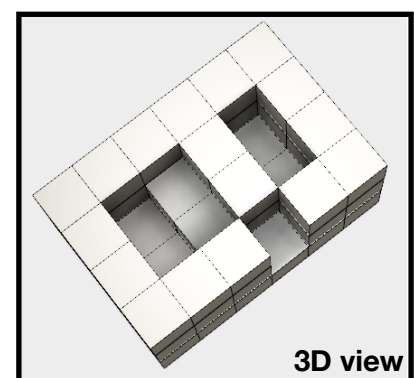
The 1 area in the middle stores 1 unit of water. The 1 area on the sides will retain no water as the water will spill out. Hence this structure will hold 1 unit of water II.

Example III:

A 6x4 matrix like the following

```
[3 . 3 . 3 . 3 . 3 . 3]  
[3 . 1 . 2 . 3 . 1 . 3]  
[3 . 1 . 2 . 3 . 1 . 3]  
[3 . 3 . 3 . 1 . 3 . 3]
```

Can be represented as the 3D view shown. The second column stores 4 units, the third column stores 2 units and the fifth column stores 4 units of water. This matrix stores 10 units of water in total.

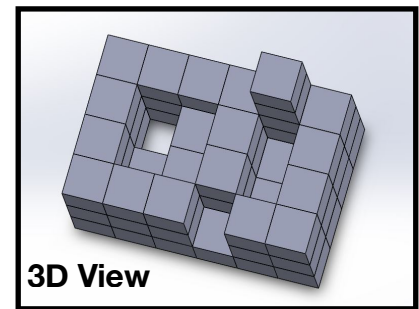


Example IV:

A 6x4 matrix like the following

```
[3 . 3 . 3 . 3 . 5 . 3]  
[3 . 0 . 2 . 3 . 1 . 3]  
[3 . 1 . 2 . 3 . 1 . 3]  
[3 . 3 . 3 . 1 . 3 . 3]
```

Can be represented as the 3D shown. This matrix resembles the example as the previous example. However, one of the elements on the second column has a 0 element. Since the structure is floating in space, the water poured would be drained out from this element.



Hence the structure won't be able to store any units of water on either the second or third columns. The only column water can be stores in is the fifth column which holds 4 units of water. Therefore this matrix holds 4 units of water in total.

Example V:

A 5x5 matrix as the following

```
[ 5 . 5 . 5 . 5 . 5]
[ 9 . 1 . 1 . 1 . 5]
[ 5 . 1 . 5 . 1 . 5]
[ 5 . 1 . 1 . 1 . 5]
[ 5 . 5 . 5 . 5 . 5]
```

Will hold 32 units of water in total, as the second column will hold 12 units, the third column will hold 8 units and the fourth column will hold 12 units, which adds up to 32 units in total.