
12. Treasure Hunt

Program Name: `Treasure.java`

Input File: `treasure.dat`

Jack has found a treasure map but only has a limited amount of time to pick up the treasure. Write a program that takes a treasure map and determines the maximum treasure that can be gathered in a given number of moves.

- The map will consist of integers between 0 and 9 inclusive. These numbers indicate the amount of treasure at the corresponding location.
- Jack can take a given number of steps. Jack can move North, South, East, or West only. Each step moves Jack one spot.
- Jack's initial position will be shown with an asterisk (*) on the edge of the map. It is possible Jack starts on a corner and so he has to take 2 steps to get onto the portion of the map that can contain treasure.
- Jack does not need to end up where he started. He can end anywhere on the map or on the boundary of the map.
- When Jack enters a location he picks up the treasure there. If he steps into that location again there isn't any more treasure at that location.

For a given map, starting location, and number of steps print out the maximum treasure Jack can collect.

Input

- The first line will contain a single integer `n` that indicates the number of data sets that follow.
- The first line in each data set will be an integer `steps` indicating the number of steps Jack can take.
- The second line of a data set will be two integers, `row` and `column`. These indicate the number of rows and columns in the treasure map not including the boundaries. All maps will be rectangular.
- The next `row + 2` lines will be the treasure map and the boundaries of the map. Each line will contain `column + 2` characters. The boundaries of the map will be indicated with periods (.), except for Jack's initial position which will be marked with an asterisk (*).

Output

For each data set print out maximum treasure Jack can collect on the map based given his starting location and the maximum number of steps

Example Input File

```
2
5
3 6
.....
.000900.
*100000.
.161101.
.....
8
5 7
.....*
.9000112.
.0102045.
.0701020.
.8412302.
.3332221.
.....
```

Example Output to Screen

```
10
18
```