UNIVERSITY CUP

ENTELECT CHALLENGE 2020

INTRODUCTION

A new alternative fuel company needs a solution for laying out their batteries within their current power grids.

They've provided the layout for some of their grids. Batteries can be placed anywhere within the grid except for a few occupied spaces for reserved components.

The issue they are facing is the placement of batteries. Power transmission and distribution is negatively affected by gaps between batteries, the best solutions would therefore have the least empty space.

PROBLEM DESCRIPTION

Your goal is to create and code an algorithm to place the batteries. Keep in mind that due to the amount of available space and size of the batteries, it may not be possible to fill every available slot.

Should there be empty space left that can't be filled, contiguous empty slots are preferred.

The batteries you will be working with are self-orientating, this removes the need for the positive and negative terminals to be placed in any specific direction.

The battery shapes are polyominos of different orders and can be rotated at 90 degrees in either direction, flipping or reflecting the shape is not permitted.

https://en.wikipedia.org/wiki/Polyomino

You will be scored based on each correctly placed battery; points will be deducted for empty spaces relative to the size of the whitespace. Additionally, you will gain a score modifier which is based on the ratio between the total capacity of the batteries used and the total area that the batteries occupy, which you want to maximise.

GRIDS

The grids that you will be tasked will placing batteries are quadrilateral in shape and described by having R rows and C columns. Within each grid there are spots that already have B components placed in them. [0,0] is situated on the top left corner of the grid.

BATTERY SHAPES

The batteries come in 25 different shapes and sizes, outlined in the shape index below. To make things easier, we have provided an additional document describing each shape within a bounding box. This will make it easier to represent each shape programmatically and is provided more as a suggested method of representing the shapes available for use. Do NOT be bound by the representations provided in the json file.

Each map will use a subset of the full list of shapes provided, and the quantities of each shape will be specified in the map files.

Shape Representation

The shapes outlined in the index below is just a single view of the how it can be used, since all 90-degree rotations are valid to be used.

ID – This corresponds to the identifier for a specific shape listed in the index and will be used as reference in the map files.

Capacity – This defines the storage capacity of each battery shape.

Shape – Visual representation of the shape.

INPUT FILE

Limits

0 < Rows < 10000

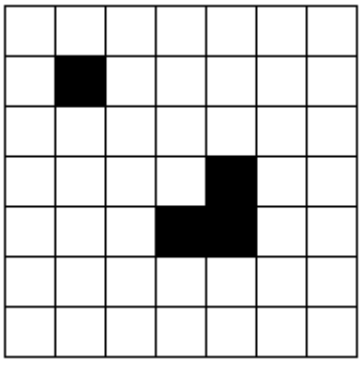
0 < Columns < 10000

0 < Reserved spaces < 10000

0 < Number of unique shapes < 25

Example: Given a 7x7 grid, fill in the batteries (black blocks are reserved).

Input	Description
7,7	Dimensions of grid [R,C]
8	Number of unique shapes available
4	Number of blocked cells in grid
1,4	Shape id 1, available 4
2,1	Shape id 2, available 1
3,3	Shape id 3, available 3
10,2	Shape id 10, available 2
11,3	Shape Id 11, available 3
15,1	Shape id 15, available 1
16,2	Shape id 16, available 2
24,1	Shape id 24, available 1
1,1|3,4|4,3|4,4	Cells [1,1], [3,4], [4,3], [4,4] blocked



Visually the grid will look like this.

SUBMISSIONS

Submissions will need to be made on the Entelect Challenge portal https://challenge.entelect.co.za/university. Each submission will need to consist of a solution output file containing information described below as well as a zipped file containing the source code used to generate your solution.
You may upload multiple submissions per map, the best scoring submission will be used.
You must complete as many maps as possible, to have the best chance at achieving a high score.

OUTPUT FILE FORMAT

For each map provide a solution file labeled in the format

Each line in a submission file should contain the coordinates and shape id being placed into the grid, separated by a pipe '|', with the coordinates [r,c] separated by a comma.

EXAMPLE output.txt:

15|0,0|0,1|1,0|1,2|2,0|2,1	Shape 15, placed at [0,0] [0,1] [1,0] [1,2] [2,0] [2,1]
1|0,2|0,3|1,3	Shape 1, placed at [0,2] [0,3] [1,3]
11|0,4|0,5|0,6	Shape 11, placed at [0,4] [0,5] [0,6]
10|1,4|1,5|1,6|2,4|2,6	Shape 10, placed at [1,4] [1,5] [1,6] [2,4] [2,6]
3|3,0|4,0|4,1|5,1	Shape 3, placed at [3,0] [4,0] [4,1] [5,1]
16|2,2|2,3|3,1|3,2|3,3|4,2	Shape 16, placed at [2,2] [2,3] [3,1] [3,2] [3,3] [4,2]
3|2,5|3,5|3,6|4,6	Shape 3, placed at [2,5] [3,5] [3,6] [4,6]
10|5,0|5,2|6,0|6,1|6,2	Shape 10, placed at [5,0] [5,2] [6,0] [6,1] [6,2]
1|5,3|6,3|6,4	Shape 1, placed at [5,3] [6,3] [6,4]
16|4,5|5,4|5,5|5,6|6,5|6,6	Shape 16, placed at [4,5] [5,4] [5,5] [5,6] [6,5] [6,6]

For a submission to be valid, the contents of each file should be in the above format.

Invalid placements will be ignored for the final scoring, and only correctly places cells will be considered.

SCORING

The solution file will be processed and scored once submitted. The source code will be required for verification purposes and must be included in the upload.

Each team's goal is to get the highest overall score possible.

You will gain points for each correctly placed piece and lose points for incorrectly placed pieces (this could be due to batteries overlapping each other or reserved spots, as well as batteries placed out of bounds).

You gain points for placing a battery cell within the grid, and the score received is aligned with the amount of space that gets filled up.

You will lose points for each empty space, if these empty spaces are separated you will lose points by the multiple of the number of empty slots across your ending grid. It is best to optimize your solution and keep empty slots next to each other.

Gaining Points:

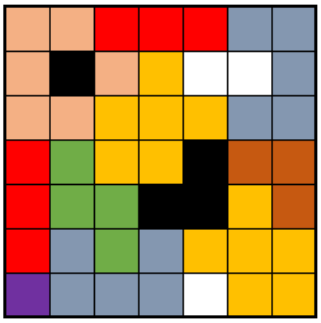
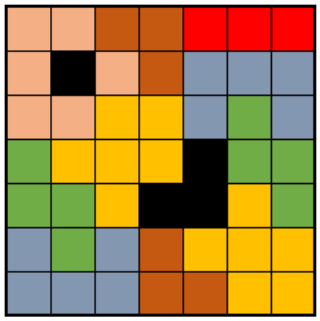
- 10 points for each space filled

Losing Points:

- 4 points for every single whitespace, not connected to other white spaces
- OR, number of cells in a group of connected whitespaces multiplied by 2

Score modifier

- Two values are generated from the solution
  - Total capacity (total_capacity)
  - Total number of spaces filled by the shapes being placed (total_area_filled)
- The ratio of total_capacity/total_area_filled is multiplied to your points after gaining and losing points.
- The final score will be rounded up to the nearest digit.

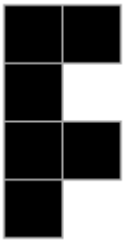
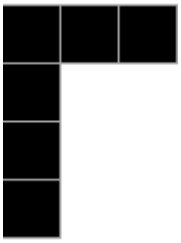
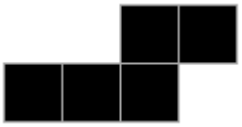
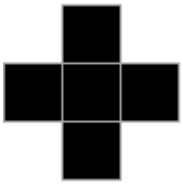
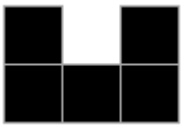
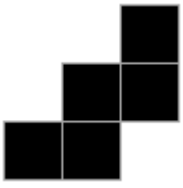
Example Visual Solutions	Score Calculation
	Total Points = 42*10 - 4 - 2*2 = 412 Score Modifier = (10+4+7+10+4+6+5+10)/42 = 56/42 = 1.3333 Final Score = 412 * 1.333 = 549.333 = 550
	Total Points = 45*10 = 450 Score Modifier = (10+5+4+7+6+10+6+7+5+10)/45 = (70)/45 = 1.5555 Final Score = 450*1.5555 = 700

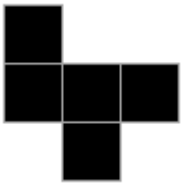
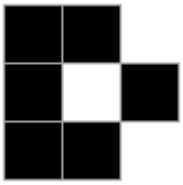
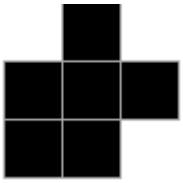
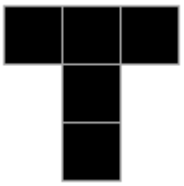
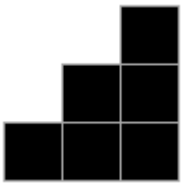
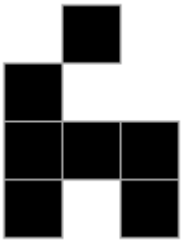
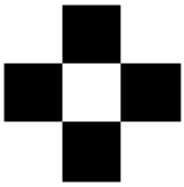
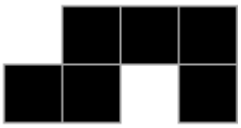
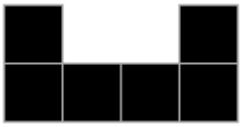
IF YOU HAVE ANY QUESTIONS OR COMMENTS, PLEASE ASK THE TEAM VIA EMAIL, FORUM OR DISCORD:

challenge@entelect.co.za

https://forum.entelect.co.za

https://discord.gg/PyA6C2R

ID	Capacity	Shape	ID	Capacity	Shape
1	5		2	6	
3	6		4	9	
5	9		6	3	
7	6		8	7	
9	7		10	7	
11	4		12	9	

13	8		14	7	
15	10		16	10	
17	5		18	7	
19	9		20	12	
21	8		22	8	
23	8		24	2	
25	12