



Level 3

Glados:

Oh, my goodness, you are fast. I've checked your records while you were asleep - I could hardly believe them, but they were right. Wait, can a robot really *believe* something?

Wait, I digress - we'll have to start coming up with routes on this new planet's surface...

We'll send a few of our more ... rudimentary scavenging robots at first

Task for Level 3:

**Starting from a point on the planet and
always choosing the shortest path see where you might end up**





Level 3

- › The world is the same as described in level 2
- › Let's assume you are put in a position with coordinates **(i, j)** (i=row, j=column)
- › Our robot scavengers can only go 3 ways:
(i, j+1), (i+1, j), (i-1, j)
- › where a +1 or -1 would mean going one cell in the respective direction
- › The scavenger will **always follow the path of least resistance**, meaning that, it will always choose the one cell which is nearest to its current one (still, from the 3 choices above)
- › In case of **equality** order by **lowest row** and then **lowest column**
- › They can also not pass through an already visited position
- › You will have to output the cell coordinates of the path it follows

Input format:

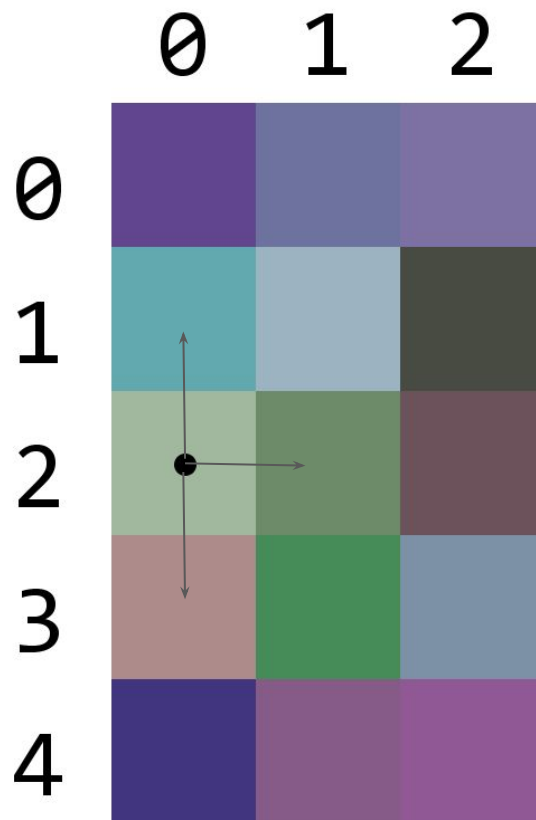
<NrRows> <NrColumns>

<StartRow> <StartColumn>

<R> <G> ... <R> <G>

...

<R> <G> ... <R> <G>



*Click to see animated version



Example input:

5 3

2 0

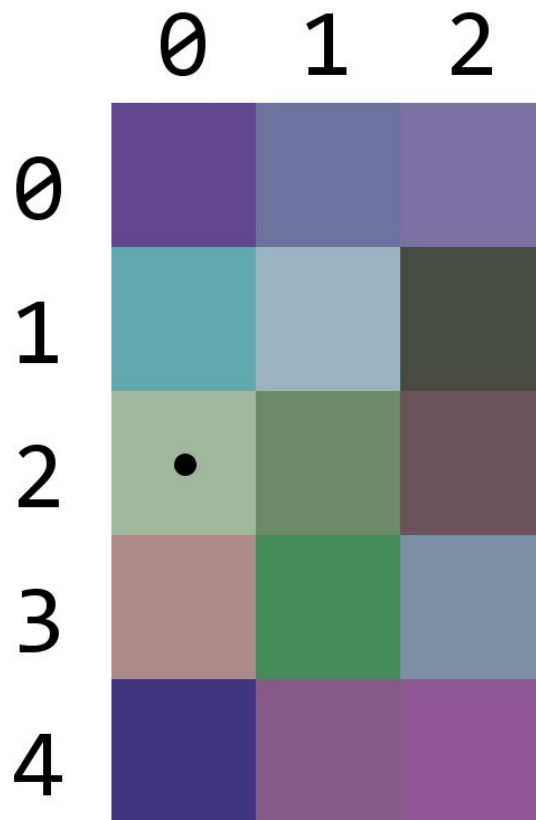
97 69 142 109 114 158 125 113 164

97 168 175 156 179 194 72 75 66

162 184 158 109 139 104 108 82 90

173 139 139 70 140 88 124 144 166

64 53 126 134 91 136 144 89 149

[*Click to see animated version](#)

Output format:

<Row> <Column>

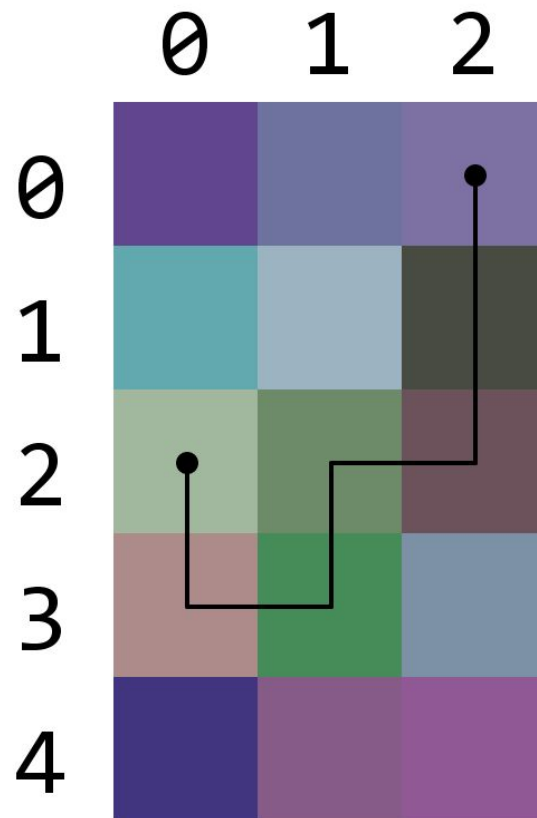
...

<Row> <Column>

} for each point of the path

Example:

```
2 0
3 0
3 1
2 1
2 2
1 2
0 2
```



*Click to see animated version