Day 6

A group of n friends is planning to watch a movie and they want to locate a contiguous seating arrangement in the same row. The seating layout of the movie theater can be visualized as a two-dimensional matrix, where vacant seats are denoted by 0s and occupied seats are denoted by 1s.

```
[[1, 0, 0, 0, 1, 1, 1],
[1, 1, 1, 0, 1, 1, 1],
[1, 0, 1, 0, 1, 0, 1],
[1, 1, 0, 1, 1, 0, 1],
[1, 0, 1, 1, 1, 1, 1],
[1, 0, 1, 1, 0, 0, 0]]
```

Develop a function that takes a seating arrangement and the number of friends (n) as inputs and outputs the count of available seating options where all n friends can sit together. In the provided scenario, if n equals 3, there would be two available spots for seating (the first row and the last row).

Examples

```
groupSeats([
  [1, 0, 1, 0, 1, 0, 1],
  [0, 1, 0, 1, 0, 1, 0],
  [0, 0, 1, 1, 1, 1, 1],
  [1, 0, 1, 1, 0, 0, 1],
  [1, 1, 1, 0, 1, 0, 1],
  [0, 1, 1, 1, 1, 0, 0]
], 2) → 3

groupSeats([
  [1, 0, 1, 0, 1, 0, 1],
  [0, 1, 0, 0, 0, 0, 0],
], 4) → 2
```

Even if multiple free arrangements overlap, they should still be considered as separate and distinct arrangements (refer to example #2 for clarification).