

**Problem Statement** — You're looking to move into a new apartment, and you're given a list of contiguous blocks on that street where each block contains an apartment that you could move into block where each block contains some services that it offers. In order to pick your apartment, you want to optimize its location in such a way that the maximum distance to any services that you care for is minimized.

Sample Input blocks = [{"gym": false, "school": true, "store": false}, {"gym": true, "school": false, "store": false}, {"gym": true, "school": true, "store": false}, {"gym": false, "school": true, "store": false}, {"gym": false, "school": true, "store": true}]

reqs = ["gym", "school", "store"]

I have Solved this question in two different languages python and JavaScript.

Code Repository: [GitHub Link](#)

### **Detailed explanation of above code:**

1. The **blocks** array is an array of objects, where each object represents a block and has three properties: gym, school, and store. Each property is a boolean value indicating whether the block has the corresponding amenity (e.g. gym is true if the block has a gym, false otherwise).
2. The **reqs** array is an array of strings, where each string represents an amenity that is required (e.g. "gym", "school", "store").
3. The **distancebetween** function calculates the absolute difference between two numbers. It takes in two arguments: a and b, and returns the absolute difference between them (i.e. Math.abs(a - b)).

4. The **minValueIndex** function finds the minimum value in an array and returns its index. It takes in an array A and does the following: The **blocks** array is an array of objects, where each object represents a block and has properties for various amenities (such as "gym", "school", and "store"). The value of each property is a boolean indicating whether the block has the corresponding amenity.
5. The **reqs** array is an array of strings representing the amenities that are required.
6. The **minValueIndex** function finds the minimum value in an array and returns its index. It takes in an array A and does the following:
  - Initializes the variables minimum to the maximum possible value (Number.POSITIVE\_INFINITY) and minIndex to 0.
  - Loops through each element in the array A.
  - If the current element is smaller than minimum, sets minimum to the current element and minIndex to the current index.
  - After the loop, returns minIndex.
7. The **apartmentHunting** function takes in two arguments: blocks and reqs. It does the following:
  - Initializes the maxDistanceForEveryBlock array to have the same length as the blocks array, and fills it with the maximum possible negative value (Number.NEGATIVE\_INFINITY).
  - Loops through each block in the blocks array.
  - For each required amenity (as specified in the reqs array), calculates the distance from the current block to the nearest block that has the required amenity.
  - Finds the maximum distance to a required amenity for the current block, and stores it in the maxDistanceForEveryBlock array at the same index as the current block.
  - After the loop, calls the minValueIndex function with the maxDistanceForEveryBlock array as an argument, and returns the result.

8. Finally, the code exports the result of the **apartmentHunting** function as a property of the exports object, with the name apartmentHunting. This would allow other modules to import and use the result of the **apartmentHunting** function.