+calculateReductionInDetection(detectionPercent, distanceA, distanceB): double

**View Map**

The computer will display a map of all the cities, the user can enter the name of a city to get description of the city along with a cost of how much time/money it will take to move there which will be determined by the distance from their current location and how many times they have previously relocated. A time cost based on distance will also be calculated along with a reduction of detection % and displayed to the user. The computer will then prompt them to enter another city name. If an invalid city name is selected an error message will appear. The user may type ‘Q’ to quit and the game menu will be displayed. Users cannot go back to previous cities or they will be caught by the FBI.

An equation will be used to calculate the cost of the relocation: ((Distance of city A + Distance of city B) \* 100) + (# of times moved \* 100) = Cost

An equation will be used to calculate the reduction in detection %: Detection % - ((Distance of city A + Distance of city B) \* 5) = New Detection %

Here is the task, inputs, output and validation rules for the +calculateReductionInDetection() function.

Task

Calculate the reduction in detection.

Inputs

detectionPercent: The percentage of detection risk.

distanceA: The distance of city A.

distanceB: The distance of city B.

Output

The new detection risk percentage: newDetectionPercent.

Validation Rules

* The distanceA and distanceB variables need to be greater than 0.
* The detectionPercent starts at 0; it cannot go below zero and cannot go above 100. When it reaches 100, the game is over.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| +calculateReductionInDetection() **Test Matrix** |  |  |  |  |  |
|  | Test Cases |  |  |  |  |
|  | Valid | Invalid | Invalid | Boundary | Boundary |
|  | **1** | **2** | **3** | **4** | **5** |
| **Inputs** |  |  |  |  |  |
| detectionPercent | 100 | 30 | 50 | 1 | 50 |
| distanceA | 5 | 4 | 5000 | 20000 | 10000 |
| distanceB | 6 | 5 |  |  |  |
|  |  |  |  |  |  |
| **Output** |  |  |  |  |  |
| newDetectionPercent | 52 | -1 | -1 | 3 | 51 |
| **Error Msg.** | No msg. | You were detected. | Amount Paid is too low | No msg. | No msg. |

+calculateReductionInDetection(detectionPercent, distanceA, distanceB): double

BEGIN

IF (distanceA < 0 || distanceB < 0) THEN

RETURN -1

IF (detectionPercent < 1) THEN

Return -1

newDetectionPercent = detectionPercent – ((distanceA + distanceB) \* 5)

RETURN newDetectionPercent

END