**Calculate the train's speed (end user story)**

The plays now are outside of the building and realize that it will be dark soon. He decides to uses another transportation to be very fast somewhere else. He has only 10 minutes until dark and the enemies become more actives. The next secure place is 1500 meters ahead and his option is to use a little train that moves very fast. When he gets to the points, he finds this table:

Desire distance in meters:

Time spent: 30 s

Acceleration: 7m/s

The initial speed in m/s:

Once he knows the desire distance, insert the initial velocity necessary to get the next secure point in safe?

This is the formula

v0 = (d / t) - [(a \* t) / 2]

* **v0** means "initial velocity"
* **d** means "distance"
* **a** means "acceleration"
* **t** means "time"

v0= (1500/30) – [(7\*30)/2]

v0= 50 – 105

v0= 55m/s

**Task: Calculate** de train´s speed in m/s

**Inputs:**

Distance: The distance in meters.

Time Spent: In seconds.

Acceleration: In seconds

**Output:** The initial speed in m/s

**Validation Rules:**

The distance can´t be different than 1500.

The time spent can´t be different than 30.

The acceleration can´t be different than 7.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | |  | |
| calcTrainSpeed Test Matrix | | | | | | | |
|  | Test Cases | | | | | | |
| Valid | Invalid | | | Boundary | | |
| 1 | 2 | 3 | 4 | 5 | 6 | |
| Inputs | | | | | | | |
| Distance | 1500 | 1600 | 1500 | 1500 | 1500 |  | |
| Time | 30s | 30s | 15s | 30s | 30s |  | |
| Acceleration | 7s | 7s | 7s | 8s | 7s |  | |
| Outputs | | | | | | | |
| Initial Speed | -55 m/s | -1 | -1 | -1 | -55 m/s |  | |
| Error |  | Distance is greater than 1500 | time is less than 30 | Acceleration greater than 7 |  |  | |

**Algorithm**

calcTrainSpeed(distance,InitialSpeed): double

BEGIN

IF (distance <> 1500) THEN

RETURN -1

IF (time <> 30) THEN

RETURN -1

IF (acceleration <> 7) THEN

RETURN-1

IF (initialSpeed) <> -55) THEN

RETURN -1

initialSpeed = (distance/time) – [(acceleration \* time) / 2];

RETURN initialSpeed

END