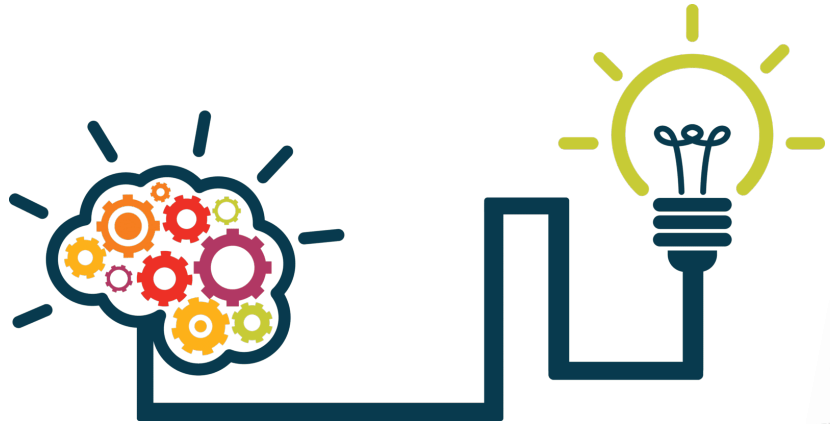



MATHS


ARTS


CSS
PHP
JS


Herbs, Spices, Healthy Oils
Grains
Proteins
Colorful Fruits and Vegetables

NO (0) YES (1)

cheese

ham

tomato

red onion

mayonnaise

argula

Code for  ?

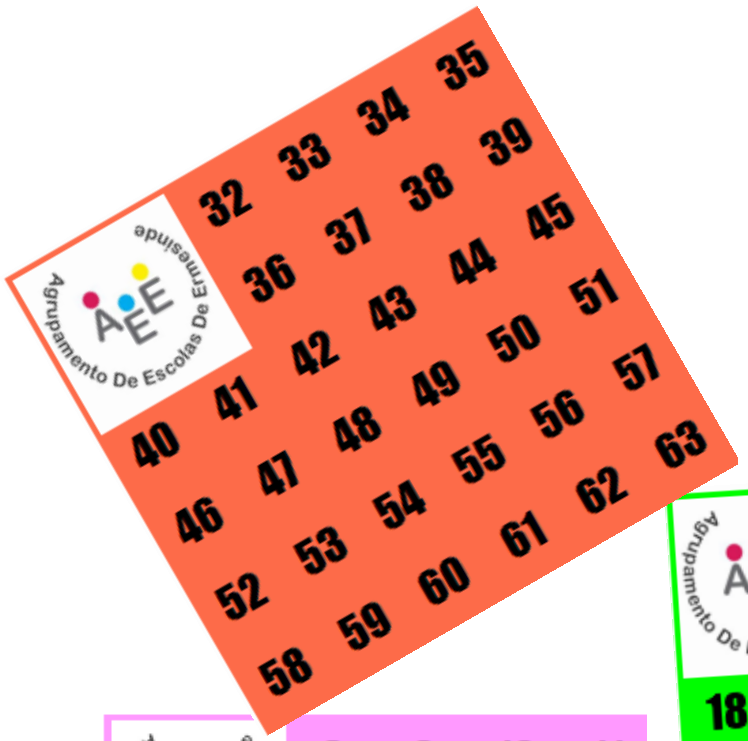
What kind of sandwich for the code **0 1 1 0 1 0** ?

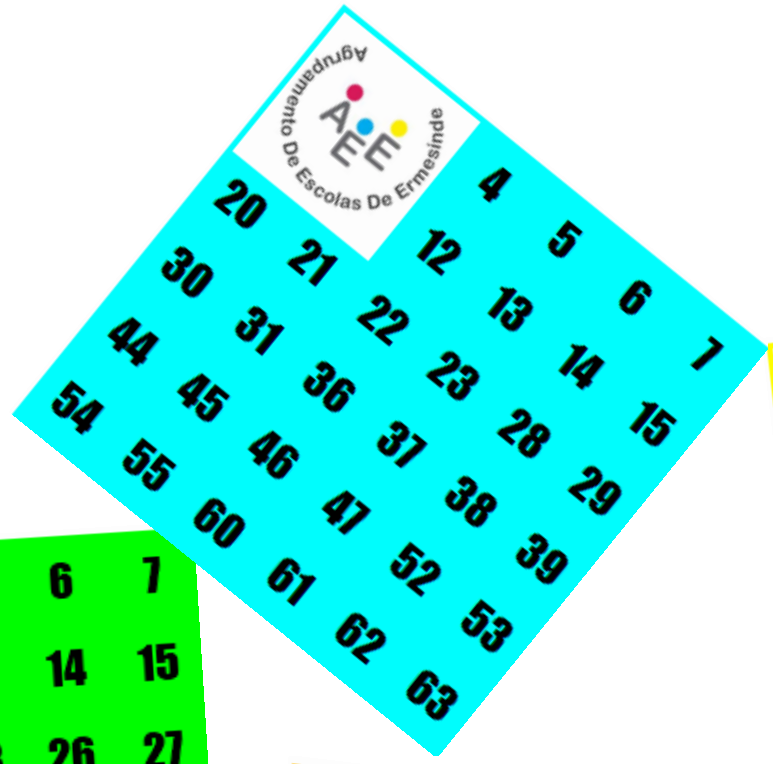
How many different sandwiches?

How many different sandwiches with cheese and without mayonnaise?

1 1 1 1 1 1


?
1 - 63


Agrupamento De Escolas De Ermesinde
32 33 34 35
36 37 38 39
40 41 42 43 44 45
46 47 48 49 50 51
52 53 54 55 56 57
58 59 60 61 62 63


Agrupamento De Escolas De Ermesinde
4 5 6 7
12 13 14 15
20 21 22 23
30 31 36 37 28 29
44 45 46 47 38 39
54 55 60 61 52 53
62 63


Agrupamento De Escolas De Ermesinde
1 3 5 7
9 11 13 15
17 19 21 23 25 27
29 31 33 35 37 39
41 43 45 47 49 51
53 55 57 59 61 63


Agrupamento De Escolas De Ermesinde
8 9 10 11
12 13 14 15
24 25 26 27 28 29
30 31 40 41 42 43
44 45 46 47 56 57
58 59 60 61 62 63


Agrupamento De Escolas De Ermesinde
2 3 6 7
10 11 14 15
18 19 22 23 26 27
30 31 34 35 38 39
42 43 46 47 50 51
54 55 58 59 62 63


Agrupamento De Escolas De Ermesinde
16 17 18 19
20 21 22 23
24 25 26 27 28 29
30 31 48 49 50 51
52 53 54 55 56 57
58 59 60 61 62 63

1	0	0	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	0	0	1	0	0
5	0	0	0	1	0	1
6	0	0	0	1	1	0
7	0	0	0	1	1	1
8	0	0	1	0	0	0
9	0	0	1	0	0	1
10	0	0	1	0	1	0
11	0	0	1	0	1	1
12	0	0	1	1	0	0
13	0	0	1	1	0	1
14	0	0	1	1	1	0
15	0	0	1	1	1	1
16	0	1	0	0	0	0
17	0	1	0	0	0	1
18	0	1	0	0	1	0
19	0	1	0	0	1	1
20	0	1	0	1	0	0
21	0	1	0	1	0	1
22	0	1	0	1	1	0
23	0	1	0	1	1	1
24	0	1	1	0	0	0
25	0	1	1	0	0	1
26	0	1	1	0	1	0
27	0	1	1	0	1	1
28	0	1	1	1	0	0
29	0	1	1	1	0	1
30	0	1	1	1	1	0
31	0	1	1	1	1	1
32	1	0	0	0	0	0
33	1	0	0	0	0	1
34	1	0	0	0	1	0
35	1	0	0	0	1	1
36	1	0	0	1	0	0
37	1	0	0	1	0	1
38	1	0	0	1	1	0
39	1	0	0	1	1	1
40	1	0	1	0	0	0
41	1	0	1	0	0	1
42	1	0	1	0	1	0
43	1	0	1	0	1	1
44	1	0	1	1	0	0
45	1	0	1	1	0	1
46	1	0	1	1	1	0
47	1	0	1	1	1	1
48	1	1	0	0	0	0
49	1	1	0	0	0	1
50	1	1	0	0	1	0
51	1	1	0	0	1	1
52	1	1	0	1	0	0
53	1	1	0	1	0	1
54	1	1	0	1	1	0
55	1	1	0	1	1	1
56	1	1	1	0	0	0
57	1	1	1	0	0	1
58	1	1	1	0	1	0
59	1	1	1	0	1	1
60	1	1	1	1	0	0
61	1	1	1	1	0	1
62	1	1	1	1	1	0
63	1	1	1	1	1	1

		32	33	34	35	
		36	37	38	39	
40	41	42	43	44	45	
46	47	48	49	50	51	
52	53	54	55	56	57	
58	59	60	61	62	63	

		16	17	18	19	
		20	21	22	23	
24	25	26	27	28	29	
30	31	48	49	50	51	
52	53	54	55	56	57	
58	59	60	61	62	63	

		8	9	10	11	
		12	13	14	15	
24	25	26	27	28	29	
30	31	40	41	42	43	
44	45	46	47	56	57	
58	59	60	61	62	63	

		4	5	6	7	
		12	13	14	15	
20	21	22	23	28	29	
30	31	36	37	38	39	
44	45	46	47	52	53	
54	55	60	61	62	63	

		2	3	6	7	
		10	11	14	15	
18	19	22	23	26	27	
30	31	34	35	38	39	
42	43	46	47	50	51	
54	55	58	59	62	63	

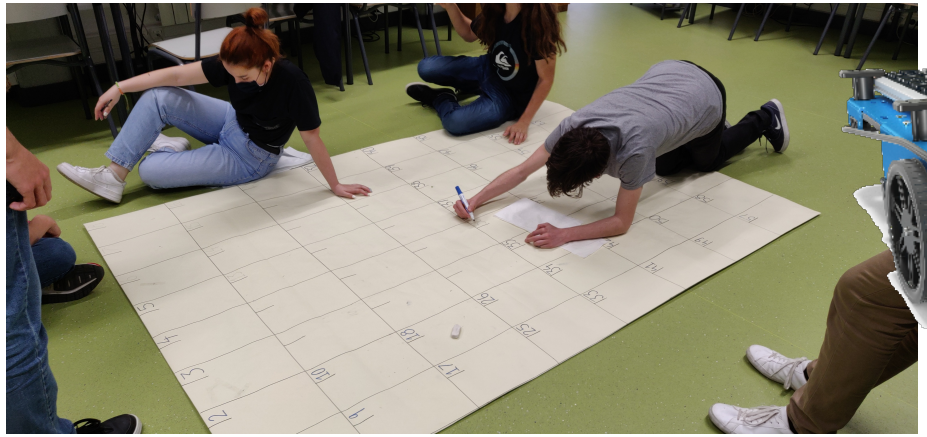
[Sem título]

		1	3	5	7	
		9	11	13	15	
17	19	21	23	25	27	
29	31	33	35	37	39	
41	43	45	47	49	51	
53	55	57	59	61	63	

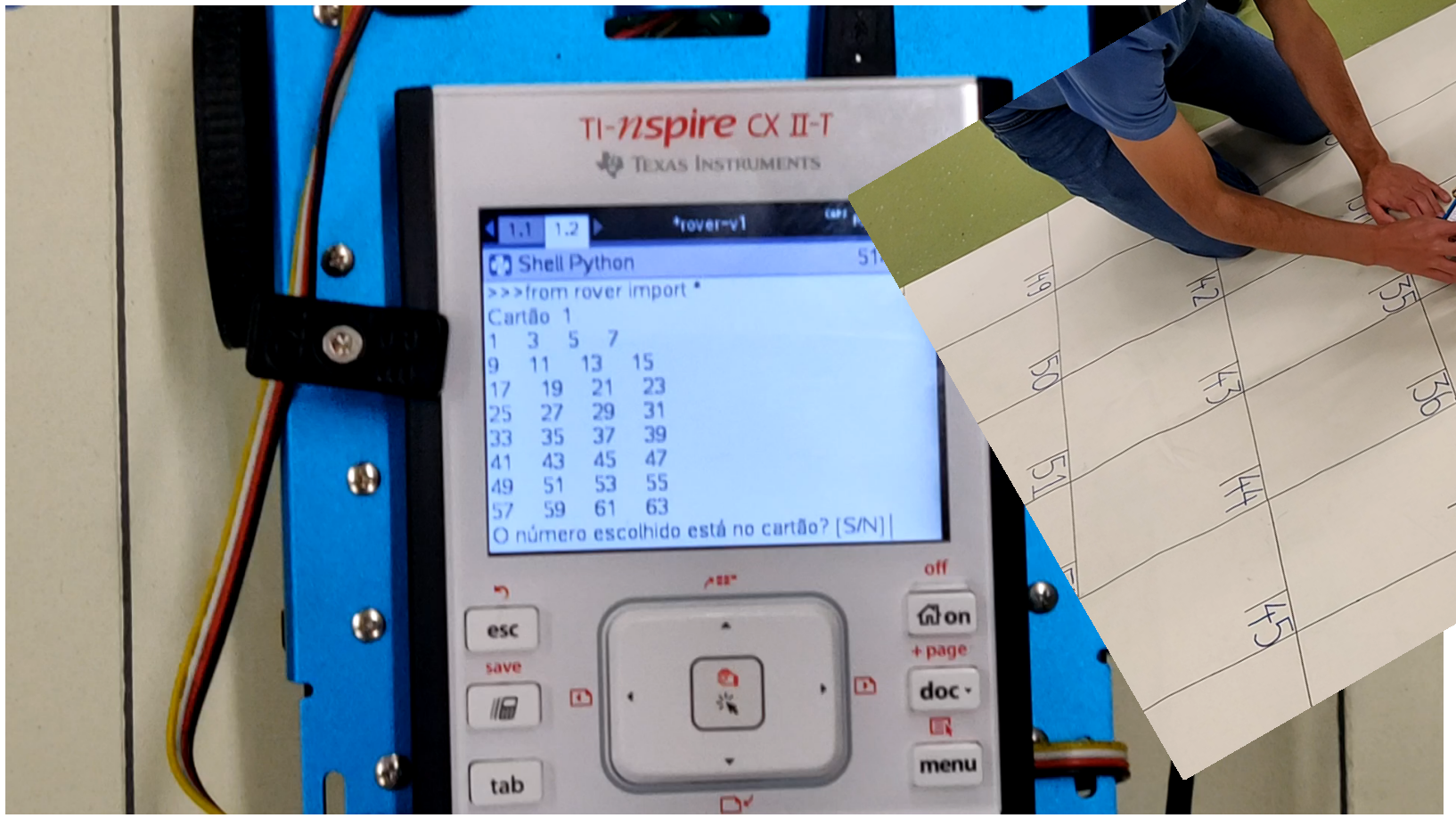
Go further!

Less cards, more cards, same base, other base,…

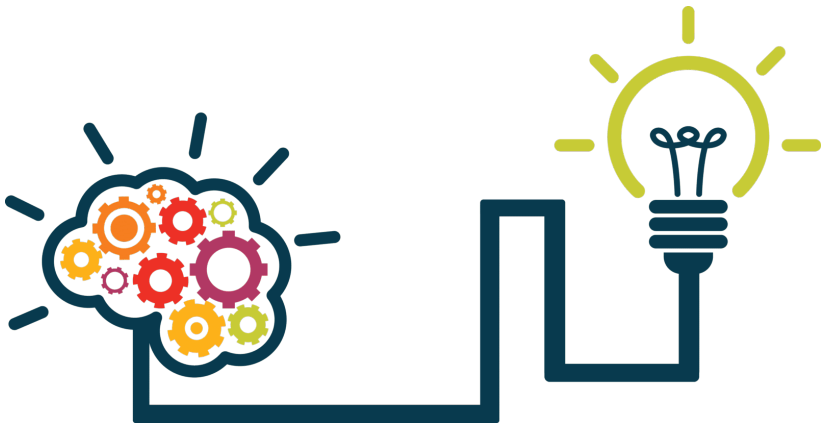
… making magic with a robotic car and Python programming…




TI-Innovator Rover


TI-nspire CX II-T
Texas Instruments
1.1 1.2
*rover-v1
Shell Python
>>>from rover import *
Cartão 1
1 3 5 7
9 11 13 15
17 19 21 23
25 27 29 31
33 35 37 39
41 43 45 47
49 51 53 55
57 59 61 63
O número escolhido está no cartão? [S/N]
esc
save
tab
off
on
+page
doc
menu


CSS
PHP
JS




MATHS


STATS
15
10
5
-20
-10
10
20
30
40
50
60

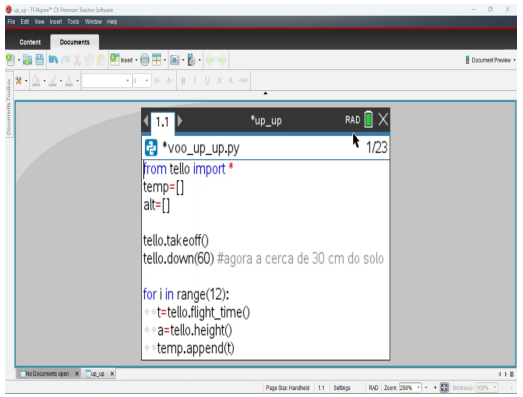
Consider the experiment that consists of collecting and recording information regarding the drone's flight time, since takeoff, as well as the height at which it is.

This data will be collected by the drone at eight different times, with a regularity that depends on the time of positioning from one record to another, as a response to a Python program, such as the one in the figure on the side.

The program's sequence of instructions leads to the following actions:

```
from tello import *
temp=[]
alt=[]

tello.takeoff()

tello.down(60)
#now, from about 30 cm of the ground

for i in range(7):
    t=tello.flight_time()
    a=tello.height()
    temp.append(t)
    alt.append(a)
    tello.up(20)
t=tello.flight_time()
a=tello.height()
temp.append(t)
alt.append(a)

store_list("tempos",temp)
store_list("alturas",alt)

tello.land()
```

1. import from the Tello module to Python;
2. take off;
3. lower 60 centimeters (to stay about 30 centimeters from the ground);
4. Repeat these actions seven times:
   a) collect data (time – since taking off and height)
   b) record the data in lists;
   c) climb 20 centimeters; (except after the 6th registration)
5. Again, a) and b), but not c).
6. record the data in lists;
7. land on the ground;
8. Save the data in lists on a worksheet.



Realize the flying experience with TI-Nspire technology and the Tello drone. (something like in the video)

```
from tello import *
temp=[]
alt=[]

tello.takeoff()

tello.down(60)
#agora a cerca de 30 cm do solo

for i in range(7):
    t=tello.flight_time()
    a=tello.height()
    temp.append(t)
    alt.append(a)
    tello.up(20)
t=tello.flight_time()
a=tello.height()
temp.append(t)
alt.append(a)

store_list("tempos",temp)
store_list("alturas",alt)

tello.land()
```

QUESTIONS

Fill in the table next to it with the data from the experiment and put it in a spreadsheet.

Times	Heights

- What, in this context, is the meaning of the first values of each of the lists?
- How long did it take from the time the first record was made to the last record?

You may observe at least one inconsistency between these data and the instruction of the successive 20-centimeter climb that the program defines.

- Estimate an appropriate value for the percentage of error in the records of this drone, relative to the height, based on the data from this experiment. Please provide possible reasons for this.

In the worksheet, add two more columns, one with the index of the record, a kind of order of the record, but starting at 0 and ending at 5 (6 records), and another with the times, but only from the first record. Note that this second column can be obtained from the time column by subtracting the time from the first cell of the column (*timesdiff=time-times[1]*).

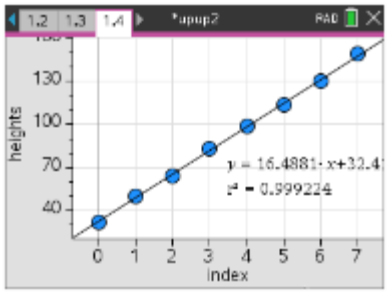
Times	Heights	index	Timesdif
		0	
		1	
		2	
		3	
		4	
		5	
		6	
		7	

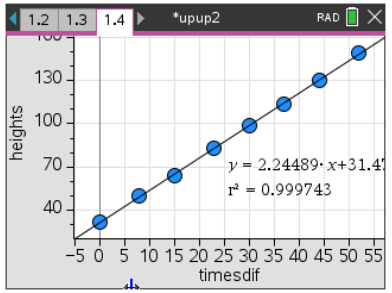
```
from tello import *
temp=[]
alt=[]

tello.takeoff()

tello.down(60)
#now, from about 30 cm of the ground

for i in range(7):
    t=tello.flight_time()
    a=tello.height()
    temp.append(t)
    alt.append(a)
    tello.up(20)
t=tello.flight_time()
a=tello.height()
temp.append(t)
alt.append(a)

store_list("tempos",temp)
store_list("alturas",alt)

tello.land()
```

- Represent each of the point clouds, (index, heights) and (timesdif, heights) and, for each, obtain the regression line and its equation.

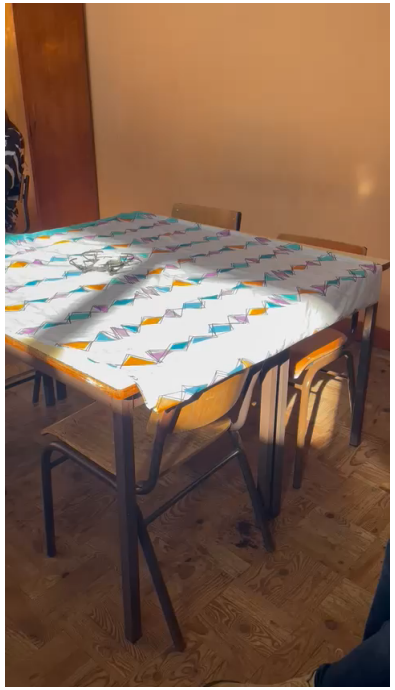
Consider the parameters of the respective equations, rounded to tenths, and record the information you have obtained.

- Equation of the line for (index, heights) $r1: y = \_\_\_\_\_x + \_\_\_\_\_$
- Equation of the line for (timesdif, heights) $r2: y = \_\_\_\_\_x + \_\_\_\_\_$





- Explain, in context, the values of the parameters in the two regression lines, coefficient of and independent term x
- How long does the drone take, on average, between each pair of records? Display the result in seconds, approximate to tenths. Explain your answer!
- If 15 records were made, could the experiment be done in a room where the ceiling was 2.7 meters from where the drone takes flight? Explain your answer!

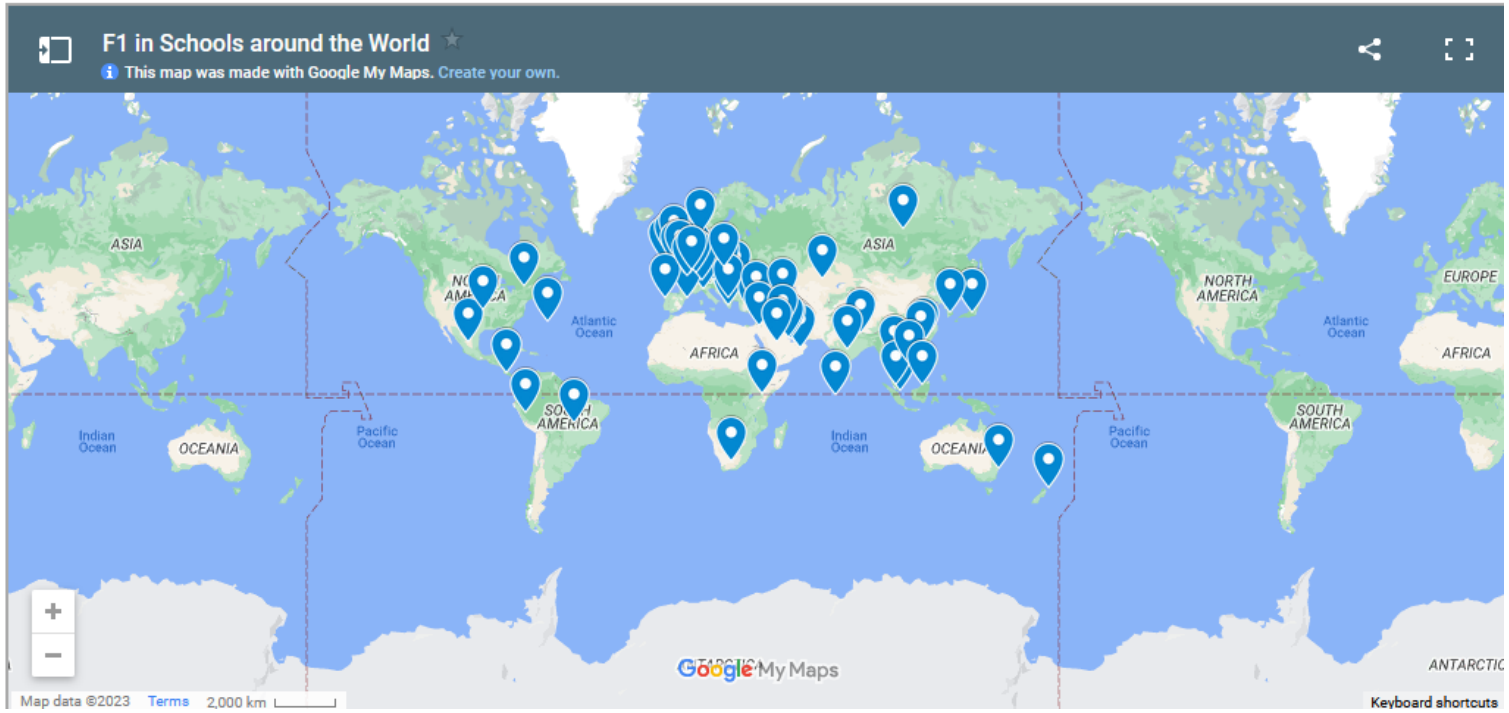


Consider the changes to the program that you can see on the side.

```
tello.down(55)

for i in range(7):
    t=tello.flight_time()
    a=tello.height()
    temp.append(t)
    alt.append(a)
    tello.up(24)
```

- Provide a prevision for each of the regression lines seen before, considering that the percentage of error in the heights between records is maintained and that the drone takes the same time between the first and the last record. Explain your predictions!

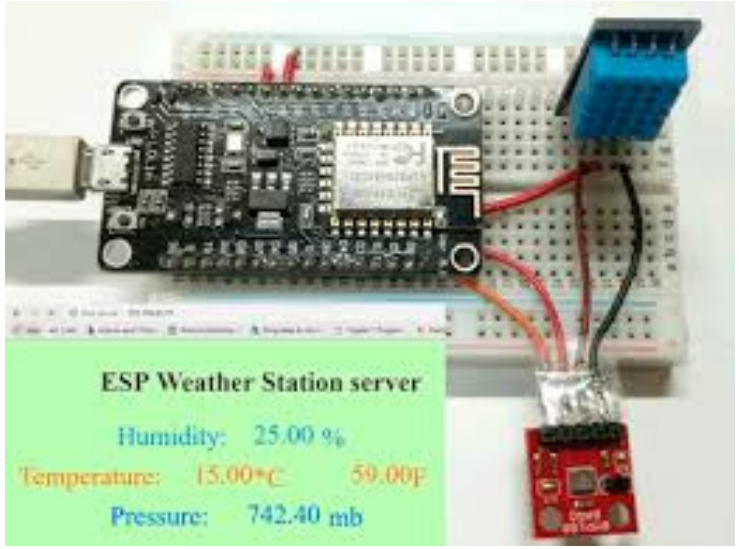
F1 in Schools is an international STEM (science, technology, engineering, mathematics) competition for school children (aged 11–19), in which groups of 3–6 students have to design and manufacture a miniature car out of the official F1 Model Block using CAD/CAM design tools.



F1® in Schools

Scarecrow with weather station

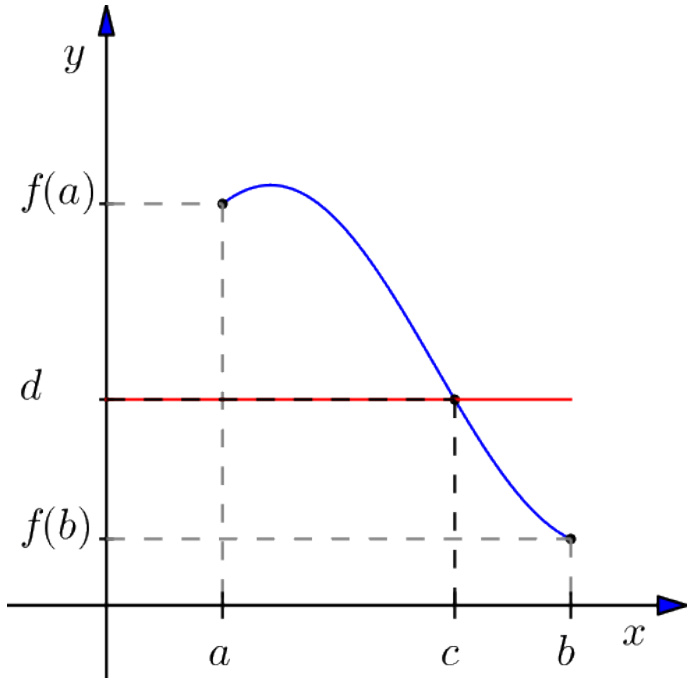
As part of the Living Science Club / Programming and Robotics Club the students created a scarecrow to scare away birds and equipped it with a set of weather sensors



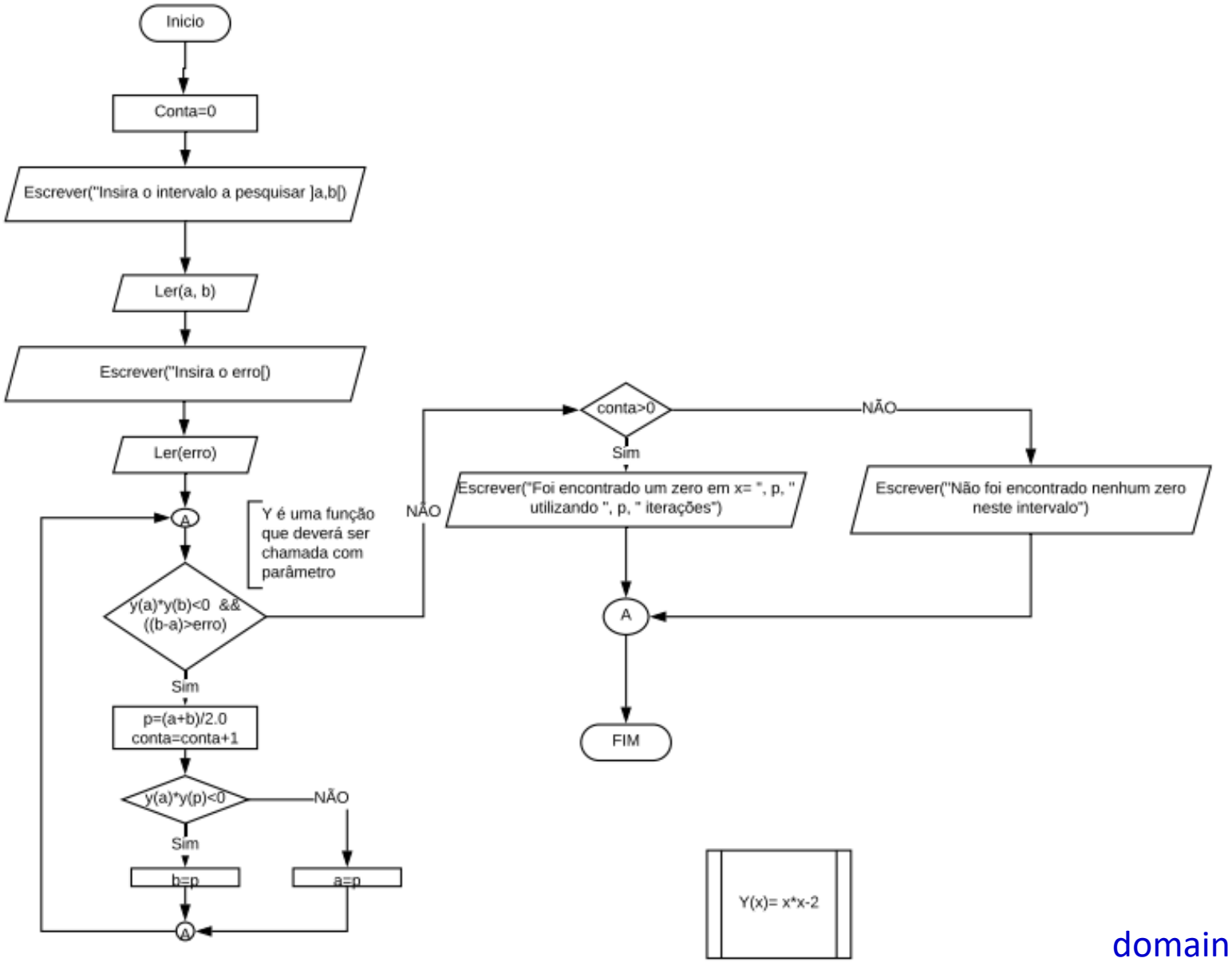
Bolzano-Cauchy Theorem

Integrating mathematics and computer content in the development of an algorithm and program for solving the Bolzano Theorem

The statement of **Bolzano's theorem**, also known as the Intermediate Value Theorem or the Bolzano–Cauchy theorem is as follows: If ***f*** is a continuous function on a given interval **[a,b]**, then for any value ***d*** between ***f(a)*** and ***f(b)***, there exists at least one value ***c*** between ***a*** and ***b*** such that ***f(c) = d***. But after all, what is the meaning of this? I will try to explain what has been said here using a small example:



Bolzano-Cauchy Theorem



domain of curricular autonomy