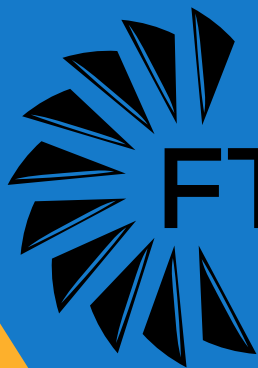




THE CRC ROBOTICS
SENIOR PROGRAMMING PROBLEMS
PRELIMINARY 1

MODUEL
2026



presented by

FTAI AVIATION

A program of

**AEST
EAST**

Version 1.0

A FEW NOTES

- The complete rules are in section 4 of the rulebook.
- You have until **Friday, November 14th, 11:59 pm** to submit your code.
- Feel free to use the programming forum on the CRC discord to ask questions and discuss the problem with other teams. It is there for that purpose!
- **We are giving you quick and easy-to-use template files for your code and the tests. You are required to use them.**

USING THE TEMPLATE FILE

- The template tests call the function to test, take the output and allow you to quickly check if your code works as intended. **All your code, except additional functions you create, should be written in the function of the part of the problem you are solving.**
- Points given in the document are indications of how difficult the section is and how many points you will get if you complete it. This preliminary problem is going to be 2% of the main challenge towards the global score of the programming competition and for more points related information consult the rulebook.

STRUCTURE

Every problem contains a small introduction like this about the basics of the problem and what is required to solve it.

Input and output specification:

Contains the inputs and their format, and which outputs the code is required to produce and in what format they shall be.

Sample input and output:

Contains a sample input, sometimes containing sub examples in the sample input, and what your program should return as an output.

Explanation of the first output:

Explains briefly the logic that was used to reach the first output, given the first input.

Platypus!

I don't believe platypuses are real animals. They are straight out of a dream, or some would say, a nightmare. No simple and accurate description can be made of this animal. Although a simple description is not possible, we'll explore a few characteristics of this peculiar animal from Australia. You'll probably be surprised at some of its characteristics.

Part 1: Draw the platypus' tail (15 points)

In Québec, we have beaver tails, but do you think they have platypus' tails in Australia? Let's try to imagine what this classic Québec dessert would look like in Oceania. In order to satisfy every client, we must have many sizes of platypus' tails available. You will have to take as input the sizes of tails in the client's order and then draw them to the console with the chars "|", "_", ".", "\", "/" and " " (space) while following these rules:

- the height of the large part = $n+1$
- the height of the diagonal part = n
- the width of the base = n
- the interior of the tail will be filled with this pattern of dots and underscores " _."

N.B. watch out when printing the backslash ("\") it might give errors, but make sure tests are working, that's the only thing that matters.

Input and output specification:

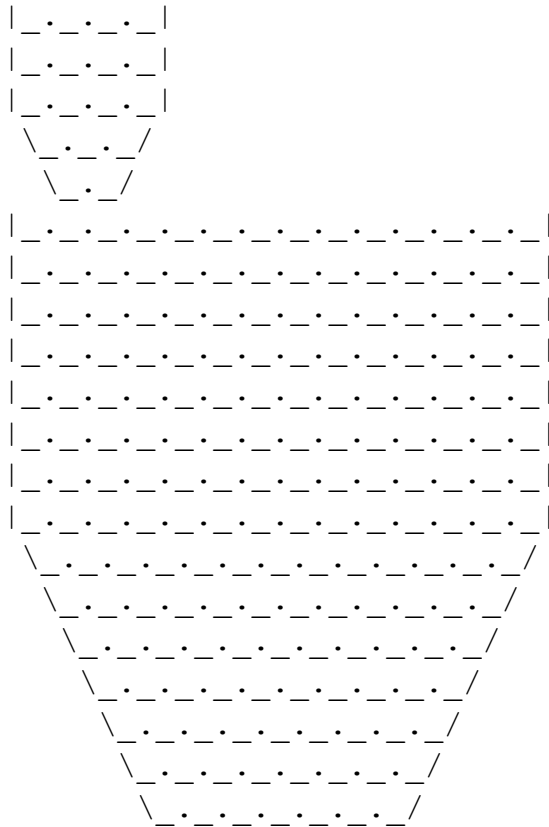
As input, you will receive a list with the size of every platypus' tail you will have to draw in the console. As output, you will have to give a list of strings to draw the tails using the rules set above.

Sample input:

[3, 2, 7]

Sample output:

```
| _ . _ . _ . _ . _ |
| _ . _ . _ . _ . _ |
| _ . _ . _ . _ . _ |
| _ . _ . _ . _ . _ |
 \ . _ . _ . _ . _ /
  \ . _ . _ . _ . _ /
   \ . _ . _ . _ . _ /
```



Explanation of the first output:

To start, let's take the list we were given as input and find the size of the first tail we need to draw, which is $n = 3$. To continue, we need to find out what the width of the widest part of the tail should be. The base of the tail's fill pattern should contain 3 underscores and the diagonal part should be 3 lines high. Since we add one underscore to every line in the diagonal part and the first line of the widest part, we get a fill pattern of $3 + 3 + 1 = 7$ underscores for the vertical part of the tail. We can now start drawing our tail. The vertical part of the tail will be constructed out of $n + 1$ ($3 + 1 = 4$) lines with an infill pattern with 7 underscores as follows: "`|_._._._._|`". We can now move on to the diagonal part of the tail. Every new line will need to start and end with one more space than the last one. The first line of this part of the tail should start and end with a single space character as follows: " `_._._./` ". We'll continue adding spaces at the start and end of the line while making the tail part shorter until we reach the end of the tail (3 lines in this case). And voilà, we just drew our first platypus tail!

Part 2: Milk (10 points)

Did you know that the platypus gives its milk to its kids by sweating it. We will find the quantity of milk produced by the platypus. The quantity of milk is determined by the volume and the age of the animal. **The milk is stored in the lower abdomen of the platypus so only half of the platypus produce milk.**

Each year of life, the platypus produces 10% more milk starting with 10% at the age of 1, then 20% at the age of 2 and the same thing applies until the age of 10.

The platypus has an elliptical shape, so here's the equation for the volume of a 3D ellipse:

$$V = \frac{4}{3}\pi * a * b * c$$

V is the volume of the platypus

a is half of the width of the platypus

b is half of its length of the platypus

c is half of its height of the platypus

Input and output specification:

For input, you will receive :

w: an *int* of the width of the platypus

h: an *int* of the height of the platypus

l: an *int* of the length of the platypus

a: an *int* of the age of the platypus

For output, you need to output a *float* with a precision of 2 decimals

Sample input:

w = 10, h = 14, l = 50, a = 4

w = 16, h = 17, l = 63, a = 7

w = 13, h = 10, l = 47, a = 3

w = 9, h = 12, l = 40, a = 8

Sample output:

733.04

3140.34

479.88

904.78

Explanation of the first output:

The dimensions of the first platypus are 10 cm of width, 14 cm of height and 50 cm of length. This platypus is 4 years old. The volume of the platypus is calculated using half of its dimensions.

$$\begin{aligned}V &= \frac{4}{3}\pi * a * b * c \\V &= \frac{4}{3}\pi * \frac{10}{2} * \frac{14}{2} * \frac{50}{2} \\V &= 3665,191429\end{aligned}$$

We divide the volume by 2 because only the inferior part of the platypus produces milk and we get 1832,595.

Then we take the volume and multiply it by 10% for each year of life, so 40% and we round it and that gives us 733.04.

Part 3: How is it written?? (25 points)

Platypus is a strange name for an animal but it is nowhere near as strange as the version in french that is: "ornithorynque". How can you even have that word in a normal sentence? Well, if we get a bit creative, we can look for the letters that are in the word "ornithorynque" and find them individually in a sentence or text! We will need to find the position of all the letters of "ornithorynque" in a sentence and if the letters are not there, the position needs to be indicated as -1.

Input and output specification:

For input, you will receive a test in which to find the letters of "ornithorynque".

For output, you will give an array of the position of the letters within the given text. The position of the letters starts at 0 for the first letter. The blank spaces and all characters including punctuation all count as a character in the position count. Each letter from the same position can only be used once and if a letter is not found, the position given is -1.

Sample input:

"Platypus is the best animal in the world."

"Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur hendrerit nulla suscipit quoniam rhoncus luctus. Curabitur in ex est. Aenean non sollicitudin nulla"

"What exactly is that thing called an ornithorynque in french??"

Sample output:

[36, 37, 22, 9, 3, 13, -1, -1, 4, 29, -1, 6, 14]

[1, 2, 30, 6, 20, 67, 13, 16, 94, 48, 92, 9, 3]

[37, 38, 24, 13, 3, 1, 43, 44, 11, 35, 47, 48, 5]

Explanation of the first output:

In the first example, we have the sentence "Platypus is the best animal in the world." We can take each letter from "ornithorynque" and try to find it in the original sentence. The first appearance of an "o" is at the position 36. For the letter "r" it is at position 37. We can also see more easily for the letter "t" that it appears at the 3rd position ("p" is 0, "l" is 1, "a" is 2 and "t" is 3).

The second time we have the letters "o" and "r", they are not found since they appear only once and were already used.

Finally, the letter "q" does not appear in the original text so the position is set to -1.

Part 4: Time to hunt! (20 points)

Did you know that, in addition to having a unique appearance, the platypus also has a real superpower? Thanks to electrolocation, it can detect the faint electrical fields produced by the muscles of aquatic insects and crustaceans it feeds on. Let's put ourselves in the shoes of a platypus for a moment and go hunting!

Imagine our platypus swimming in an infinitely large pool of water, filled with delicious prey. Starting from its initial position, your goal will be to find the closest prey and sort them by distance, from the nearest to the farthest. That way, our platypus will know which prey to head towards in order to save as much energy as possible during its hunt!

Input and output specification:

For input, you will receive serie of coordinates in 3D from the pool. The first coordinate is the one of the platypus whilst the other one are the coordinates of the preys.

For output, you need to provide an array of int that will organize the preys in ascending order of their distance with the platypus.

Helper: Here's the formula to compute the distance between 2 coordinates in 3 dimensions

$$D = \sqrt{\Delta a^2 + \Delta b^2 + \Delta c^2}$$

Sample input:

[(0, 0, 0), (5, 6, 4), (9, 9, 2), (6, 2, 3), (3, 8, 4)]

[(0, 0, 0), (4, 13, 6), (2, 6, 4), (3, 1, 7), (7, 5, 10)]

[(16, 17, 5), (4, 4, 9), (11, 16, 14), (8, 4, 9), (11, 18, 8), (17, 2, 15), (10, 18, 7)]

Sample output:

[3, 1, 4, 2]

[2, 3, 4, 1]

[4, 6, 2, 3, 5, 1]

Explanation of the first output:

To find the ranking of the distances of the preys, we first have to compute the distance for all the preys. Here's the computation for the first one:

$$\begin{aligned} D_1 &= \sqrt{\Delta a^2 + \Delta b^2 + \Delta c^2} \\ D_1 &= \sqrt{(5 - 0)^2 + (6 - 0)^2 + (4 - 0)^2} \\ D_1 &= \sqrt{25 + 36 + 16} \\ D_1 &= 8.77496 \end{aligned}$$

We can now do the same thing to find the distance between the platypus and the other prey. We get $D_2 = 12.8841$, $D_3 = 7$ and $D_4 = 9.43398$.

The closest is the 3rd prey, followed by the first one, then the 4th one and finally the furthest away is the second prey. This gives us the following order [3, 1, 4, 2].

Part 5: Game of Platypus (35 points)

A platypus lives alone on a huge territory, and it separates its land using a 16x16 grid so it knows what it should do next.

You will be given a 16x16 grid through a 2D array. Each element is a char, which represents either the platypus ('x'), some food ('.') or an empty space ('_'). For every round, the platypus can move 1 block horizontally or vertically (i.e. they can move up, down, left, or right) if some food exists in that block. If multiple food exists in those places, the platypus will always prioritize the order right > down > left > up. If there is no food in those places, it will try to find the closest food (depending on the Manhattan distance) and move towards it. The platypus will always move horizontally (left/right) first and then vertically (up/down). If it can't find something to eat for 3 consecutive rounds, it will die.

Now, your mission is to determine whether the platypus will stay alive after k rounds. Print 'no' if the platypus is no longer alive after at most k rounds, and print 'yes' if the platypus stays alive after k rounds.

Hint:

The Manhattan distance between two points (x_1, y_1) and (x_2, y_2) is defined as:

$$d = |x_1 - x_2| + |y_1 - y_2|$$

Input and output specification:

You will first receive a variable of type *int*, representing the integer K ($10 \leq K \leq 25$) which is the amount of turns to simulate.

.

You will also receive an array of 16 *strings* with 16 characters each, all the characters be either "x", ".", or "_". Representing the initial placement of platypus, food, and empty space.

You will need to output the *string* "Yes" if the platypus is still alive after k rounds, or "No" otherwise.

Sample Inputs:

turns: 11

```
__x_ . _____ . -
_____ . _____
_____ . . ____ . -
_ . __ . _____ . ____ .
_____ . ____ . ____
. _____ . . .
_ . _____ . ____
__ . _____ . . ____ .
_____ . _____ . ____
. ____ . ____ . _____
_____ . _____
_____ . . . ____ . ____
__ . . ____ . . ____
_____ .
_____ . ____ . ____
_____ . . ____ . ____
```

turns: 18

```
__ . _____ . ____
_____
_____ .
_____ . ____ . . ____ . .
_____ . ____ . . ____
_ . _____ x
_ . _____
_____ . _____
_____
. _____ . _____
_____
_____ . ____ . _____
_____
_____ . ____
_____ .
_____ . ____ . _____
```

Sample Outputs:

Yes

No

Explanation of the first output:

=====

INITIAL BOARD STATE

=====

Platypus starting position: (0, 3)

Total food on board: 41

0123456789012345

0	__	x	.	_____	.	_
1	_____	.	_____			
2	_____	.	.	__	.	_
3	_	.	__	.	_____	.
4	_____	.	__	.	_____	
5	.	_____	.	__	.	
6	_	.	_____	.	_____	
7	__	.	_____	.	.	__
8	_____	.	_____	.	_____	
9	.	__	.	__	.	_____
10	_____	.	_____			
11	_____	.	.	.	_____	.
12	__	.	.	__	.	.
13	_____	.	_____			
14	_____	.	__	.	_____	
15	_____	.	.	_____	.	__

=====

ROUND 1/11

=====

Platypus position: (0, 3)

Starvation counter: 0/3

Remaining food: 41

✗ No adjacent food found

Closest food at (0, 5), distance: 2

Moving right from (0, 3) to (0, 4)

✗ Moved toward food but didn't reach it yet

Starvation counter: 1/3

Board after Round 1 :

0123456789012345

0	__	x	.	_____	.	_
1	_____	.	_____			
2	_____	.	.	__	.	_
3	_	.	__	.	_____	.
4	_____	.	__	.	_____	
5	.	_____	.	__	.	
6	_	.	_____	.	_____	
7	__	.	_____	.	.	__
8	_____	.	_____	.	_____	
9	.	__	.	__	.	_____
10	_____	.	_____			
11	_____	.	.	.	_____	.
12	__	.	.	__	.	.
13	_____	.	_____			
14	_____	.	__	.	_____	

15 ____..____. _

=====

ROUND 2/11

=====

Platypus position: (0, 4)

Starvation counter: 1/3

Remaining food: 41

✓ Found adjacent food to the right!

Moving from (0, 4) to (0, 5)

Eating food! (Starvation counter reset to 0)

Board after Round 2 :

0123456789012345

0 ____x____. _
1 _____. _
2 _____. _
3 _.._. _
4 _____. _
5 _.._. _
6 _.._. _
7 _.._. _
8 _____. _
9 _.._. _
10 _____. _
11 _____. _
12 _____. _
13 _____. _
14 _____. _
15 _____. _

=====

ROUND 3/11

=====

Platypus position: (0, 5)

Starvation counter: 0/3

Remaining food: 40

✗ No adjacent food found

Closest food at (1, 7), distance: 3

Moving right from (0, 5) to (0, 6)

✗ Moved toward food but didn't reach it yet

Starvation counter: 1/3

Board after Round 3 :

0123456789012345

0 ____x____. _
1 _____. _
2 _____. _
3 _.._. _
4 _____. _
5 _.._. _
6 _.._. _
7 _.._. _
8 _____. _

```

9  ._._.
10 _____
11 ____...____.
12 __..__..
13 _____
14 _____._.
15 ____..____.

```

=====

ROUND 4/11

=====

Platypus position: (0, 6)

Starvation counter: 1/3

Remaining food: 40

✗ No adjacent food found

Closest food at (1, 7), distance: 2

Moving right from (0, 6) to (0, 7)

✗ Moved toward food but didn't reach it yet

Starvation counter: 2/3

Board after Round 4 :

```

0123456789012345
0  _____x_____.
1  _____
2  _____..____.
3  _._._____.
4  _____._.
5  ._____.
6  _._._____.
7  _._._____.
8  _____
9  ._._.
10 _____
11 ____...____.
12 __..__..
13 _____
14 _____._.
15 ____..____.

```

=====

ROUND 5/11

=====

Platypus position: (0, 7)

Starvation counter: 2/3

Remaining food: 40

✓ Found adjacent food to the down!

Moving from (0, 7) to (1, 7)

Eating food! (Starvation counter reset to 0)

Board after Round 5 :

```

0123456789012345
0  _____
1  _____x_____.
2  _____..____.

```

```

3  _ . _ . _ . _ .
4  _ _ _ . _ . _
5  . _ _ _ _ . _
6  _ . _ _ _ . _
7  _ . _ _ _ . . _
8  _ _ . _ _ . _
9  . _ . _ . _ _
10 _ _ _ . _ _ _
11 _ _ . . . _ _ . _
12 _ . . _ _ . . _ _
13 _ _ _ _ _ .
14 _ _ _ . _ . _
15 _ _ . . _ _ _ . _

```

=====

ROUND 6/11

=====

Platypus position: (1, 7)

Starvation counter: 0/3

Remaining food: 39

✗ No adjacent food found

Closest food at (2, 9), distance: 3

Moving right from (1, 7) to (1, 8)

✗ Moved toward food but didn't reach it yet

Starvation counter: 1/3

Board after Round 6 :

```

0123456789012345
0  _ _ _ _ _ . _
1  _ _ _ x _ _ _
2  _ _ _ . . _ . _
3  _ . _ . _ . _ .
4  _ _ _ . _ . _
5  . _ _ _ _ . .
6  _ . _ _ _ . _
7  _ . _ _ _ . . _
8  _ _ . _ _ . _
9  . _ . _ . _ _
10 _ _ _ . _ _ _
11 _ _ . . . _ _ . _
12 _ . . _ _ . . _ _
13 _ _ _ _ _ .
14 _ _ _ . _ . _
15 _ _ . . _ _ _ . _

```

=====

ROUND 7/11

=====

Platypus position: (1, 8)

Starvation counter: 1/3

Remaining food: 39

✗ No adjacent food found

Closest food at (2, 9), distance: 2

Moving right from (1, 8) to (1, 9)

X Moved toward food but didn't reach it yet
Starvation counter: 2/3

Board after Round 7 :

```
0123456789012345
0  _____.-
1  _____x_____
2  _____..____.-
3  _.-_._____.____.
4  _____.-.____
5  ._____.-.
6  _.-_____.-
7  _.-_____..____.-
8  _____._____.____
9  .-_._____.____
10 _____.-
11 _____._____.____
12 _____._____.____
13 _____.-
14 _____.-.____
15 _____._____.____
```

=====

ROUND 8/11

=====

Platypus position: (1, 9)
Starvation counter: 2/3
Remaining food: 39
✓ Found adjacent food to the down!
Moving from (1, 9) to (2, 9)
Eating food! (Starvation counter reset to 0)

Board after Round 8 :

```
0123456789012345
0  _____.-
1  _____
2  _____x.____.-
3  _.-_._____.____.
4  _____.-.____
5  ._____.-.
6  _.-_____.-
7  _.-_____..____.-
8  _____._____.____
9  .-_._____.____
10 _____.-
11 _____._____.____
12 _____._____.____
13 _____.-
14 _____.-.____
15 _____._____.____
```

=====

ROUND 9/11

=====

Platypus position: (2, 9)
 Starvation counter: 0/3
 Remaining food: 38
 ✓ Found adjacent food to the right!
 Moving from (2, 9) to (2, 10)
 Eating food! (Starvation counter reset to 0)

Board after Round 9 :

```

0123456789012345
0  _____.-
1  _____
2  _____x____.-
3  _.-._._____.____.
4  _____.-.-____
5  ._____.-.-
6  _.-_____.-____
7  _.-_____..-.-____
8  _____.-____
9  .-.-.-____
10 _____.-____
11 _____...____.-____
12 _.-.-____..____
13 _____.-
14 _____.-.-____
15 _____.._____.-____

```

=====

ROUND 10/11

=====

Platypus position: (2, 10)
 Starvation counter: 0/3
 Remaining food: 37
 ✓ Found adjacent food to the down!
 Moving from (2, 10) to (3, 10)
 Eating food! (Starvation counter reset to 0)

Board after Round 10 :

```

0123456789012345
0  _____.-
1  _____
2  _____.-
3  _.-._._____x____.-
4  _____.-.-____
5  ._____.-.-
6  _.-_____.-____
7  _.-_____..-.-____
8  _____.-____
9  .-.-.-____
10 _____.-____
11 _____...____.-____
12 _.-.-____..____
13 _____.-
14 _____.-.-____
15 _____.._____.-____

```

```

=====
ROUND 11/11
=====

Platypus position: (3, 10)
Starvation counter: 0/3
Remaining food: 36
X No adjacent food found
  Closest food at (4, 9), distance: 2
  Moving left from (3, 10) to (3, 9)
  X Moved toward food but didn't reach it yet
  Starvation counter: 1/3

Board after Round 11 :
  0123456789012345
0  _____.-
1  _____
2  _____.-
3  _.-.-.x____.-
4  _____.-.-
5  .-_____.-.-
6  _.-_____.-
7  _.-_____.-.-
8  _____.-_____.-
9  _.-.-.-_____-
10 _____.-_____-
11 _____.-.-_____.-
12 _.-.-.-.-_____-
13 _____.-
14 _____.-.-_____-
15 _____.-.-_____.-

=====
PLATYPUS SURVIVED!
=====

The platypus survived all 11 rounds!
Final position: (3, 9)
Food remaining: 36

=====
FINAL ANSWER
=====
Yes

```