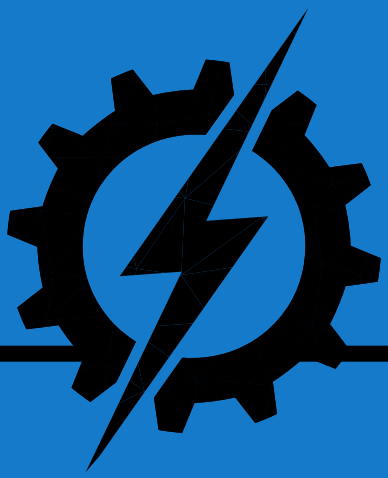
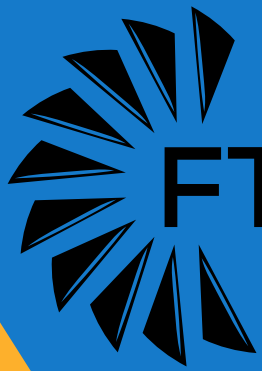


THE CRC ROBOTICS SENIOR PROGRAMMING PROBLEMS

SPECIAL CHALLENGE

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, February 13th, 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 4% 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.

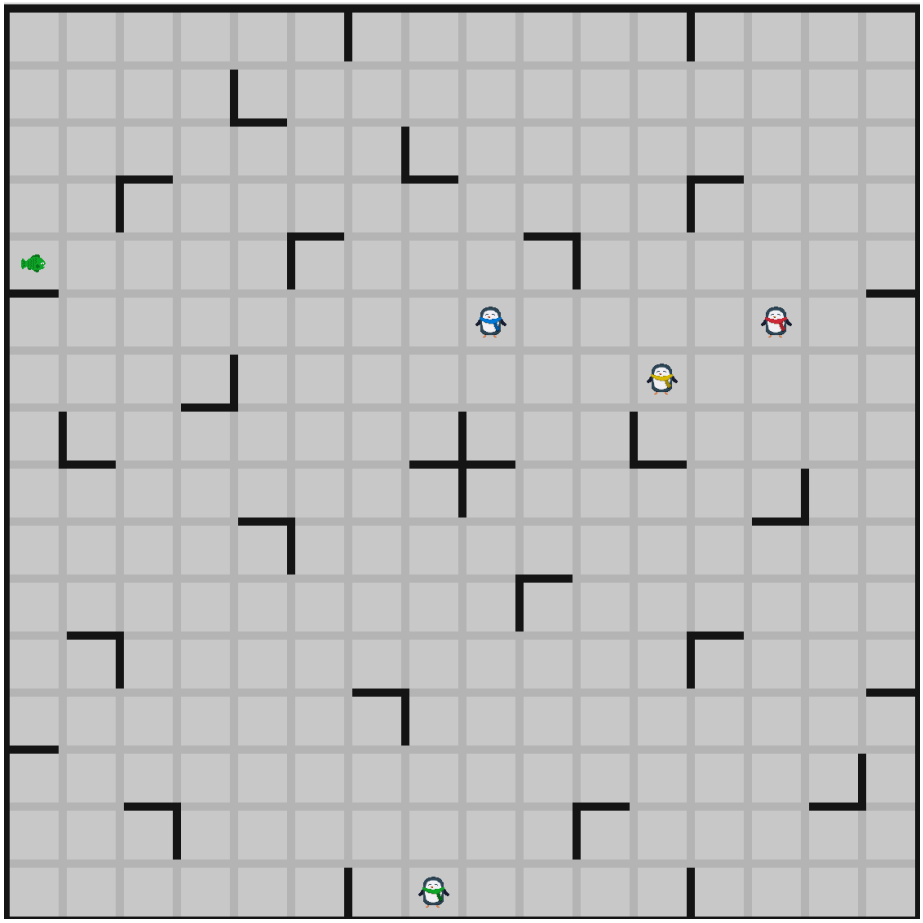
Slip N' Slide!

In this special problem, you will have to control 4 penguins on ice to catch a fish! The fish can only be caught by a penguin of a specified color. The penguins will have to work as a team to get to the fish as fast as possible!

Since the penguins are on ice, when they move, they slide until they hit a wall or another penguin. These physics apply for all 4 penguins in their quest to reach the fish.

The ice with its wall will be of dimension 16x16 and will be the same layout for all the challenge configurations. **There is only one ice layout!** However, the penguins could be starting at any position on the grid and the fish will always be at the angle of 2 walls.

To solve each configuration, your program will have 10 seconds. You will have to find the sequence of moves with the least amount of steps in the least amount of time!



Input and output specifications:

Input specifications:

You will receive the starting position of the 4 penguins as a *list* of *tuples*. The tuples are the position in X (left to right from 0 to 15), the position in Y (up to down from 0 to 15) and the uppercase first letter of the color of the penguin.

You will receive the position of the fish with the color of the penguin that has to reach it. The tuple will have the X position (left to right from 0 to 15), the Y position (up to down from 0 to 15) and the color of the penguin that has to rhea the fish.

You will also receive a Board object to help you with the ice layout and all the walls. The details of the Board objects are in the file game_layout,py. **It is NOT necessary to use the provided board**. This Board is mainly set up for display purposes and may not be useful for your problem in part because you are not allowed to change any file other than prelim.py. If you want to know a bit more, the board's grid is 33x33 to allow for the walls already organized for visualization. **It is strongly advised that you make you own representation of the ice and all the walls since it is the same one that will be used for all challenge configurations!**

Output specifications:

As an output, you will need to provide a series of commands in a list of strings. Each command is comprised of 2 elements. The first element is the uppercase letter of the color of the penguin to move and the second uppercase letter is the direction to move. (Up = U, Down = D, Right = R, Left = L)

For example, if the Red penguin needs to slide down, we have the command "RD". The solution is only a list of commands to make sure the right penguin reaches the fish!

Scoring

The scoring is separated in 2 aspects which are the precision score (amount of moves to solve the challenge) and the speed score (time used by your program to solve the challenge). For each challenge configuration, a maximum of 50 points will be awarded for the precision score and 20 points for the speed score for a total of 70 points per example. If your program does not return a valid solution within the 10 seconds allowed, the score for this example will be 0 for that configuration. Your program will be tested on the same 20 grids for all schools to give out your final score.

Precision Score

The precision score is the amount of moves that your penguins in total have to make to solve the challenge grid. If you have the solution with the least amount of steps out of all the schools, you will receive 50 points. Every move that is more than the best solution will lose you 5 points until a score of 0 if you have more than 10 moves over the best solution.

Speed Score

The speed score measures the speed of your program compared to the other schools. You have a maximum of 20 points if you find a valid solution the fastest. You lose 1 point per ranking, so only the first 20 teams get points for the speed score (as a reference, 22 schools submitted Prelim 1). To be considered in the speed ranking for a configuration, your solution has to be valid.

Number of moves (lowest amount = X)	Precision score
X + 0 moves	50
X + 1 moves	45
X + 2 moves	40
…	…
X + 9 moves	5
X + 10 moves	0
> X + 10 moves	0

Speed compared to other schools	Score de vitesse
fastest	20
second fastest	19
third fastest	18
…	…
19th fastest	2
20th fastest	1
21st and more	0

How to test your program?

You can use pytest as with any other Prelim problem.

There are 3 variables that you can change. The first 2 are for seed. You can put a number and this will guarantee you have the same grid every time you test your program. To make sure it is used, you will have to change use_seed to True.

The 3rd variable is the number of examples in a single run.

These variables are in the file test_prelim.py.