Principles of Programming Language Lab (PCC-CS-391) Exam Question, AY:2023-2024

- **Instructions**::

  Before the start of the exam, you will be given detailed verbal instructions regarding what to do and how and when to submit your solution.

  A summary of the workflow is provided below:

  You will be given a printed problem sheet (this one), and an incomplete dune project folder.

  The provided project folder contains a `bin`, `lib` and a `test` folder with appropriate ocaml and dune files in them. Most of the files are incomplete (except the ones inside the `test` folder).

  You don't need to create any new file. You only need to complete the incomplete dune and ocaml files so that you are able to build the project `mazesolver` using `dune build` command inside the `mazesolver` folder. A brief primer on the basics of how to compile a dune project, execute the binary generated and run the test cases has already been provided to you a few days ago.

  The exact details on which functions to write, their input and return types, and the functionality they are to implement have been provided to you as detailed comments within the incomplete dune and ocaml files supplied to you. Read and re-read them thoroughly till you fully understand them. Only then plan and write your code.

  Once you are satisfied that your code is complete, and the project builds without errors, run the tests provided. For running the tests just outside the `mazesolver` project folder, you are provided with a `mat.txt` file containing a simple matrix for testing. DO NOT modify or change the location of this file (otherwise the provided tests won't run).

  If the provided tests run successfully, you may write and run your own tests for checking the correctness of your code. But before submitting, remove your tests, as during evaluation, we shall be using a different test set.

- **Submission Instructions**::

  1. The sooner you submit a correct solution, higher the chance of getting better marks (subject to degree of correctness of your solution)

  2. Once you are ready to submit, create a folder named `<your_registration_number>-<your_name>`, put your completed project folder (`mazesolver`) inside it, and mail the folder (as a `zip` file) to **lostcentury720@gmail.com**, with the subject line: "PPL Lab Exam Submission from: <Your Full Name> (<Registration Number>)".

     For example, if your Registration number is 12345 and your name is James Bond, you should create the folder `12345-James_Bond`, put your completed project folder (`mazesolver`) inside this folder, along with the `mat.txt` test input file provided to you. Then make a zip file out of this folder as `12345-James_Bond.zip` and mail it to the above mentioned email address with the subject line "PPL Lab Exam Submission from: James Bond (12345)".

     In case your email service provider (for example, google) does not allow you to attach the zip file, it's most likely due to some executable present in your project folder. In that case, simply do a `dune clean` inside your project folder and then make the zip file and attach in email as mentioned above.

3. Time: Deadline for submission is exactly 6 hours from the start of your exam. Any submission after this time will not be considered for evaluation.

4. You are strongly suggested not to wait till the last moment for submitting. As mentioned above, correct code submitted earlier fetches more marks (subject to degree of correctness). And this effect of reduction of marks due to later submissions only increases as more time passes (till deadline).

- **Problem Statement**::

  In short, your program is supposed to do the following:

  1. Read from the command line 3 things:
     - the name of a text file which contains a matrix of 0s and 1s in a specified format (details provided in the material supplied to you for this exam).
     - 2 integers indicating the row and column numbers of the entry point of the matrix contained in the file read
  2. Try to find a path from the entry point to at most one exit point in the matrix, and print the exit point as a (row, column) pair.
  3. If no exit point exists, print an appropriate message indicating that.