UNIVERSITY OF TORONTO ENGINEERING KOMPETITIONS

JANUARY 15TH , 2022 - JANUARY 16TH, 2022

UTEK

COMPETITOR'S PACKAGE

PROGRAMMING

TABLE OF CONTENTS

Director: Aman Patra
Email: programming@utek.skule.ca
A special thank you to our sponsors

1.0 INTRODUCTION

Welcome to the 21st University of Toronto Engineering Kompetition! UTEK 2022 aims to provide students with an opportunity to apply class knowledge and acquire problem solving skills applicable to the real world. It is a chance to challenge oneself, meet like-minded peers, and network with company professionals. We hope you enjoy UTEK 2022!

Through this Competition, you will work together with your team to solve a programming challenge better than the competing teams and present your solution. Teams are evaluated based on cleanliness, efficiency, and functioning of their codes. We are looking forward to your participation. Good Luck!

1.1 UTEK 2022 THEME

This year, the UTEK theme is *Save earth! It's the only home we've got*. Saving the earth is a necessary step needed for our progressive future and the need for action has never been clearer. This year we hope to carry over the spirit of protection of our world while testing your problem solving skills.

The idea of saving the world can be applied in a multitude of ways. From water management and pollution control to planting trees, there are many ways to play your part in creating a clearer future for us. Today, we will focus on the aspect of Energy efficiency and conservation.

2.0 COMPETITION SCHEDULE

Time	Event
Saturday January 15th, 2022	
9:00 - 10:00 AM	opening ceremony with Vice Dean Coyle
10:00 - 10:15 AM	competition problem statement briefings
10:15 - 7:15 PM	work time
6:15 PM	submission of code due
7:15 PM	submission of presentation and output files due
Sunday January 16th, 2022	
9:30 - 10:00 AM	judge presentation debriefing
10:00 - 12:00 PM	presentations
12:00 - 1:00 PM	lunch
12:50 - 1:00 PM	judge presentation debriefing
1:00 - 3:00 PM	presentations
3:00 - 4:00 PM	awards & closing ceremony with Dean Yip
4:00 - 5:00 PM	judge feedback & networking OEC information for winning teams

3.0 RULES

- Students from all years can participate in the competition.
- You can only collaborate with the members of your team.
- You can use any programming language and its standard libraries to solve the given problem (eg. if there is a Python package needed to be installed that does Breadth First Search for you, it cannot be used. If there is a standard list sorting function, it can be used)
- Time complexity of the algorithm will be tested as well
- There can be a maximum of 4 University of Toronto students in a team.
- You are allowed to use online resources. All the online resources used by competitors must be cited in a reference slide at the end of the presentation (in whichever academic citation format they select). Competitors are not permitted to submit work completed by anyone other than the members of their team. If they decide to recycle their own or someone else's code it must be clearly cited in the presentation
- If you are not able to finish later parts of the competition, outline your solution for the presentation.

4.0 PROBLEM BACKGROUND

With the growth of autonomous machines, world production has skyrocketed to never seen heights. Industries around the world have started incorporating automated systems and now things like "robots" are more common than ever before. Now that we have reached this stage in technology, it's time we start looking at sustainability and keeping the world afloat to support our technological advances.

You and your team have recently joined the race to develop a state-of-the-art path planning algorithm for an autonomous cleaning robot at tetnofa industries. The details of the algorithm you are required to create are presented in the following sections.

Tetnofa industries has several prototype autonomous cleaning robots, each with different energy consumption specifications. Each robot will be given a list of locations that need to be cleaned and a number ranging from 1 to 100 determining the time in seconds it would require to clean the location. Movement and cleaning will both consume different amounts of energy depending on the model of the robot used. All locations are provided in cartesian coordinates with the robots starting and ending its path at its point of departure. Robots are free to traverse any path within the 100 by 100 square grid that is free of obstacles.The primary objective of tetnofa is to develop an algorithm for selecting the most efficient robot to clean certain locations and cleaning all the required locations in the most energy efficient manner. Solutions will be judged on both completeness and optimality.

5.0 INPUT OUTPUT DETAILS

Each question 1-5 has three test cases, which will be in files e.g. 1a.in, 1b.in, 1c.in. One test case (a) will be provided at the beginning of the competition, the rest will be provided one hour before the end of the competition.

The output should be in files with the same name, but extension .out, e.g. 1a.out, 1b.out, 1c.out. Make sure you submit all the files with correct naming and formatting as described below. Parts 2-5 will be auto-graded.

Part	Given Files	Submitted Files
1	1a.in, 1b.in, 1c.in	1a.out, 1b.out, 1c.out
2	2a.in, 2b.in, 2c.in	2a.out, 2b.out, 2c.out
3	3a.in, 3b.in, 3c.in	3a.out, 3b.out, 3c.out
4	4a.in, 4b.in, 4c.in	4a.out, 4b.out, 4c.out
5	5a.in, 5b.in, 5c.in	5a.out, 5b.out, 5c.out
6		Presentation

Input

The format will be as follows. Comments are provided for explanation, and will not be present in the actual input

```
2 1 1
# Three integers for the number of robots, number of locations to be
cleaned
# and number of obstacles respectively
# There will be fewer than 6 robots
swift 4 8
Heavy 8 4
# A tuple of one string and two integers for name, and respective
efficiencies # of each robot
# (name, moveEff, CleanEff)
# The efficiencies are numbers ranging from 1 to 100, specifying the
energy    # used per second used moving or per second used cleaning.
# A robot can move once per second
1 1 3
# A tuple of three values for each location to be cleaned
# (x, y, time_required)
1 2 3 5
# A tuple of four integers for each obstacle: (x1, y1, x2, y2)
# Bottom-left corner is (x1, y1), the top-right corner is (x2, y2)
# i.e. x1 <= x2, y1 <= y2
```

Output

Each robot has three possible actions, and is controlled independently: move, clean, rest. Each action takes one time unit. Lines preceded by the '#' are comments for further explanation and are not expected in the output.

```
move x y
# move to the given x,y coordinates. The robot can move to any of
the eight
# adjacent squares (like a king in chess). One second is taken for
every square # moved
# Takes two integer arguments
clean x y
# cleans the location and takes as long as specified in the input
# Takes two integer argument
rest
# Do nothing. Takes no arguments
```

The output format for parts 2-5 will be as follows.

```
# commands for the robots are separated as such
Robot swift
move 1 1
clean 1 1
move 0 0
rest

Robot zoom
rest

# first robot moves diagonally to 1, 1, cleans 1 1, moves back to
0,0 and rests
# second robot does not do any actions
```

6.0 PROBLEM DESCRIPTION

Part 1

At the end of the day, the robot is given a list of locations that it must clean as well as the time required to clean them. The locations are not aggregated. As orders come into the system, the system simply stores the results into a file. Write a program to aggregate the location and time required to clean the location.

Example Input: As described above, with no obstacles. the format is (x, y, time_required>)

```
1, 7, 0 # number_robots, number_locations, number_obstacles
alpha 4 5
5 6 3
8 9 23
5 6 3
# note that this is a duplicate, so it will only be stored once
32 49 32
42 0 69
1 1 4
1 1 5
#note that this has the same location, so only the higher time will be
stored i.e 1 1 5
```

Example Output: Aggregated output must be summarized in a file with the following format, order of elements # is that as inputted

```
Robot Name: alpha; Movement Efficiency: 4; Cleaning Efficiency: 5;
Location Number: 1; Time required: 3; Location: 5 6;
Location Number: 2; Time required: 23; Location: 8 9;
Location Number: 3; Time required: 32; Location: 32 49;
Location Number: 4; Time required: 69; Location: 42 0;
Location Number: 5; Time required: 5; Location: 1 1;
```

Part 2

The robot starts at the origin (0, 0). It will now need to clean the given locations. The robot is free to roam anywhere in the 100x100 square grid. In this part, assume there is only 1 robot given and no obstacles given.

Input: Format as given in section 5. There will only be one robot and no obstacles.
Output: Format as given in section 5.

Part 3

As before, the robot will start at location (0,0) and must go to several locations. Now, there are multiple obstacles present within a warehouse. In this part, assume there is only 1 robot given.

Input: Format as given in section 5. There will only be one robot with obstacles.
Output: Format as given in section 5.

Part 4

Tetnofa decides to buy more robots that will operate simultaneously. In this section, the robots should work together to clean the warehouse. Make sure that the robots do not collide. There are no obstacles in this section. The i'th robot will start at (i, 0). All robots must return to their location of departure.

Input: Format as given in section 5. There will be multiple robots, no obstacles.
Output: Format as given in section 5.

Part 5

Unfortunately, Tetnofa continues to have obstacles in their warehouse. Ensure that the robots do not collide with each other or with obstacles.

Input: Format as given in section 5. There will be multiple robots with obstacles.
Output: Format as given in section 5.

Part 6

Tetnofa would like to showcase their product to investors, and needs a dashboard and fancy animations to show their robots using their innovative algorithms. Your demonstration video/ screenshots should showcase something interesting about your algorithms.

Input: Use any input(s) from a previous question or make your own
Output: A video (youtube/ Drive etc. link is sufficient), gif or screenshots

7.0 COMPETITION DELIVERABLES

1. Code
All code and documentation you have written for the competition, including a README that contains relevant citations and running instructions. The code is due at 6:15.

2. Test case output
We will provide test inputs. Run your code on them and submit the output. The outputs are due at 7:15.

3. A presentation summarizing the key aspects of the problem, your approach and your solution. The slides are due at 7:15. Please note that due to space constraints, only the top 6 teams will be presenting. You will be notified by email on Saturday night.

	Time	Additional Information
Team Presentation	10 min	The Remainder will be at the 3 and 1-minute marks. There is a grace period of 30 seconds, after which the presenters **will** be cut off.
Judges Q&A	5 min	All members of the team must be ready to answer questions.

Failure to submit all the files in a correct format will result in deduction of points.

8.0 RUBRIC

Teams will be responsible for uploading their solution to each section of the problem, as well as the outputs to their programs in a file. We will use scripts that will grade each team by parsing through their outputs. Representatives from OEC will judge the presentations and look through the code. There will be a penalty for plagiarized content, absent teammates and late submissions.

Factors	Points
Corrrectness	30
Efficiency	40
Presentation	30

8.4.1 CORRECTNESS

Full points for correctness will be awarded if the code fulfills all the given conditions. The conditions to be tested for will vary with the part being checked. Points will appropriately be deducted depending on the number of failed conditions.

8.4.2 EFFICIENCY

For Parts 2-5, Efficiency or Accuracy will only be checked if code fulfills a full score of Correctness in the associated part.

$$Efficiency\ Points\ = 40\ \times \frac{Best\ Energy\ Consumption\ submitted\ for\ this\ Part}{Submitted\ Energy\ Consumption}$$

The Average of the Efficiency Points received by each of your parts will be your Final Efficiency Points.

8.4.3 PRESENTATION

The judges will award points to the presentations submitted based on level of Justification, Presentation flow or cohesiveness, and visuals or simulations.