

## Assignment #2 – Recursion: Cheapest Route

Due date: \_\_\_\_\_

Imagine that you live at the bottom-left corner of a grid (see below), and you are on your way to the movie theatre, which is at the top-right corner of the grid.

```

12 3 15 8 21
10 2 10 13 9
11 8 19 17 3

```



However, you must pay a toll at each point in the grid. The cost of each toll is represented by the numbers. You can only walk either NORTH or EAST.

Write a recursive program that will find the route to the movie theatre that will cost you a MINIMUM amount of money. (You need to save your money for the movie and popcorn!)

You will be given an input file that will resemble the following:

```

2           ← The number of grids in the test file
3           ← The # of rows in the first grid
5           ← The # of columns in the first grid
12 3 15 8 21 ← The first grid – each number is separated by a space
10 2 10 13 9
11 8 19 17 3

2           ← The # of rows in the second grid
2           ← The # of columns in the second grid
10 2       ← The second grid
15 9

```

Your output should display the route that will cost you the LEAST amount of toll money. You should display the amount of money paid at each toll in the route, the directions on how to get there, as well as the total cost. Here is the sample output to match the above file:

```

Finding the Cheapest Routes:
Grid #1:
12 3 15 8 21
10 2 10 13 9
11 8 19 17 3
Cheapest Route: 11 8 2 3 15 8 21
Direction: EAST NORTH NORTH EAST EAST EAST
Cheapest Cost: $68
Grid #2:
10 2
15 9
Cheapest Route: 15 9 2
Direction: EAST EAST NORTH
Cheapest Cost: $26
Program is Complete

```

**Here are some hints:**

- ❖ Finding the cheapest route is an extra step after you have found all the possible routes.
  - ➔ Therefore, create a program that FINDS ALL POSSIBLE ROUTES FIRST. After that is working, then find a way to keep track of the cheapest route.
- ❖ You should first decide what variables you should make GLOBAL and what you should make LOCAL. Remember, if it is local, you will need to pass it into each method.
- ❖ You should always keep track of your current route and directions, as well as keeping another set of CHEAPEST route and directions.
- ❖ Your recursive method should contain at least 3 parameters: the grid array, the current x position, and the current y position.

**BONUS:**

You've decided to move to a friendlier neighbourhood, and in this new neighbourhood, not all places collect money. In fact, some don't charge you money at all, and some even give you money for visiting their part of town! Therefore, on this new grid, these new friendlier tolls will be indicated with a zero or negative cost.

Because of this new toll layout, you will be allowed to move NORTH, EAST, SOUTH, or WEST, so that you can incur the least amount of cost. The only restriction is you cannot go to the same toll more than once. (So don't even think about going to the same place to keep on collecting money!)

---

Once again you should plan out your program carefully before beginning!

**Don't forget to COMMENT YOUR PROGRAM!!**  
(introductory, method, and internal comments)