Ben Turcotte *Assignment 2: Ada*
Bturcott - 0832217 *CIS*3190*

Algorithm:

I chose to base my algorithm on the depth-first search algorithm. The algorithm implements a stack. The stack is comprised of Coordinate (user defined type consisting of two integers: x, y) elements The steps involved are as follows:

I. push starting coordinate onto stack
II. while the stack's top element is not the end point, loop
  A. if the top element has no unvisited adjacent coordinates, pop it off the stack
  B. else, move to next available space (order of precedence: north, south, east, west)
III. done.

Developing the algorithm was the easiest part of this assignment. The depth-first-search algorithm is a very popular algorithm & it is one that I was already fairly familiar with. There were a few decisions involved.

First, I needed to decide on the order of precedence for the possible directions (north, south, east, west… what order?! oh dear there's like 24 different ways to chose from!). Yes, I realize that I was exclaiming in parentheses.

Then I needed to figure out how to deal with an unsolvable maze. That turned out not to be as difficult as I originally thought. All that was required was to make sure that the stack was never empty once it entered the loop. Then, when printing, I needed to check if the starting coordinate was the only coordinate on the stack. If it was, then the maze was unsolvable.

The difficult part of this assignment was sifting through documentation trying to figure out what to use, what not to use and why. I spent at least five times as long reading about Ada as I did actually writing code. It was especially cumbersome for file input. I wasted a lot of time trying to figure out the best way to do file input. After what I considered to be long enough, I decided that the input component of my program would just have to be a bit on the weak side. Judging by what I had read, input and output are some of Ada's fairly weak points.

What were the greatest problems faced while designing the algorithm in Ada?

Definitely file input. As mentioned above, I spent forever trying to figure out how to do that correctly.

What particular features made Ada a good language?

A lot of the syntax in Ada is fairly similar to how a person would describe a procedure in plain english. That is easily the nicest feature about Ada. The compiler is also extremely informative. Ada is probably my favourite language to debug because of that.

Would it have been easier to write the program in a language such as C?

Yes. I did. It took me half an hour.

Given your knowledge of programming, was Ada easy to learn? What structures made Ada usable? (In comparison to C for instance)

After the file input fiasco, it was fairly smooth sailing. So, I would say that, for the most part, Ada is an easy language to learn.

What did you dislike the most about Ada?

You already know what I dislike about Ada. I prefer to end this on a happy note…

:)