1994 Paper 1 Question 6

The following ML declaration introduces a data type that can be used to represent a potentially infinite arrangement of cells at positions with integer coordinates in the first quadrant of the $x - y$ plane:

```
datatype A = Z
           | Cell of int ref * A ref * A ref;
```

Each cell contains an integer value and pointers to the cells immediately to its right and above itself. These three components are all mutable so that the arrangement of the cells and the integers they contain can change during use. The constructor `Z` can be used in an `A ref` to allow a cluster to have a boundary rather than continuing through unbounded chains of cells.

Define a function `mkrow(`n`)` of type `int->A` that will return a row of length $n + 1$ cells initialised with zeros. For instance:

```
mkrow(1) = Cell(ref 0, ref(Cell(ref 0, ref Z, ref Z)), ref Z)
```

[5 marks]

Define a function `zip(`*row1*`,` *row2*`)` of type `A*A->A` that will return *row1* with *row2* joined above it. This function is entitled to change some of the `ref A` pointers in *row1*. For example

```
val root = zip(mkrow(3), mkrow(2));
```

would give `root` a value representing the following arrangement.

$$\begin{tikzcd}
 & 0 \arrow[r] & 0 \arrow[r] & 0 & \\
\texttt{root} : & 0 \arrow[r] \arrow[u] & 0 \arrow[r] \arrow[u] & 0 \arrow[r] \arrow[u] & 0
\end{tikzcd}$$

[5 marks]

Next define a function `mkarr(`m`,`n`)` of type `(int*int)->A` that will return a value representing a rectangular array of $n + 1$ rows each of which are of length $m + 1$ in which each cell is initialised to zero [5 marks]

Paths originating from the bottom leftmost cell (which will be referred to by the variable `root`) are represented by values of the type `dir list` where `dir` is declared as follows:

```
datatype dir = Right | Up;
```

Finally define a function `inc_path_cells` of type `A->A list->unit` that will increment the integers in all the cells that lie on a specified path within a given collection of cells. For instance after `root` had been set up as above, the two calls

```
    inc_path_cells root [Right, Up, Right];
    inc_path_cells root [Right, Right, Right];
```

would leave `root` representing the following arrangement:

$$\begin{tikzcd}
 & \texttt{0} \arrow[r] & \texttt{1} \arrow[r] & \texttt{1} & \\
\texttt{root :} & \texttt{2} \arrow[r] \arrow[u] & \texttt{2} \arrow[r] \arrow[u] & \texttt{1} \arrow[r] \arrow[u] & \texttt{1}
\end{tikzcd}$$

You may assume that the path does not try to reach cells outside the given arrangement. [5 marks]