Task:

Extend the programming language described by the attributed translation grammar for Tastier to include the definition and use of named scalar constants of the form:

```
const limit = 512; // example constant definition
```

and add support for the declaration and use of one and two dimensional arrays of scalar values of the form:

```
// example variable declarations
int vector[1..limit];
int matrix[1..32, 1..1024];

// examples of the use of such variables
vector[i+1] := a + b/c - d;
i := matrix[j, k+1] * 2;
```

Note:

1. A constant has a value - once set it can be accessed in an expression by name, but its value cannot be changed (hence the use of the "=" operator rather than ":=").
2. For variable arrays the bounds should be specified by integer literals (for example "1"), or by named integer constant values (for example "maxsize").

Please submit your modified version of the `Tastier.ATG` file along with a detailed report by no later than 10:00pm on Friday evening, 10th November.

In addition to submitting a detailed report along with the latest version of your `Tastier.ATG` file for coursework exercise 5, you will also need to provide the `SymTab.cs` file [with your modifications to Obj so that it can store data about constants and arrays] and the `Tastier.html` file [which should describe your modified grammar]. Finally, you should include a revised `TastierProgram.TAS` test file which demonstrates the new features you have added to the language.

As before, this exercise is more about careful analysis and design (on paper) than about coding - remember, an hour or two of analysis and design will always save several days of coding by "trial and error" on a keyboard!

Hints:

- You will need to add fields to Obj to describe a variable's “sort” (eg scalar or array) and to hold information about an array (eg the number of dimensions and their bounds).
- You need to decide how array elements will be mapped in memory and then design the ARM assembly code used to access individual elements of an array.
- Once you have decided on the syntax of your changes to the language Tastier, you might consider writing a short test program to demonstrate each of the new features in your revised version of the language and translate it by hand into ARM assembly language.
- The final step will then be the coding - ie expressing the syntax and corresponding translation in ATG form.