Topic: Information flow specification and analysis

Problem 1: Download StackPackageHW1.zip and uncompress it. Add information flow contracts (Depends aspects). Make them as precise as possible. Load the project into GPS and check that the SPARK examiner verifies that the Depends aspects are satisfied.

Problem 2: In this exercise, you will develop and use the flow matrices to demonstrate the calculation made in performing information flow analysis of the Push procedure:

procedure Push(X: in Integer) is
begin
  Top := Top + 1;
  S(Top) := X;
end Push;

To deal with the array assignment within the framework described in the lecture (spark-ifa.pdf, slides 22-42) and in the Barnes book (High Integrity Software: The SPARK Approach to Safety and Security, Addison-Wesley, 2006), we will rewrite the second assignment using the (equivalent) update notation.

procedure Push(X: in Integer) is
begin
  Top := Top + 1;
  S := S’Update(Top => X);
end Push;

For your matrices, use the following orderings.

Variables: 1) Top 2) S 3) X
Expressions: 1) Top + 1 2) S’Update(Top => X)

Following the example on slide #33, show the (extended) L, M, and P matrices for “Top := Top + 1” and “S := S’Update(Top => X)”, and show the matrix equation used to calculate the R matrix for these assignments from their L, M, and P matrices.

Then show the calculation of the R matrix for the body of Push produced using the equation $R_{S1;S2} = R_{S1} \cdot R_{S2}$

Explain how the results of this calculation justify the Depends aspect that you attached to the Push procedure.
Problem 3: In this exercise, you’ll demonstrate a few of the “interesting”
matrix calculations performed in flow analysis of the procedure Divide:

```plaintext
procedure Divide(M, N: in Natural;
Q, R, out Natural)
begin
R := M;
Q := 0;
while R <= N loop
Q := Q + 1;
R := R - N;
end loop;
end Divide;
```

In your matrices, use the following orderings:

Variables: 1) M 2) N 3) Q 4) R
Expressions: 1) M 2) 0 3) Q + 1 4) R - N 5) R >= N

Part (a): Show the R matrix for the initialization sequence — i.e., \( R_{Q_1=0; Q_i=0} \)
You do not need to do the calculation of the previous exercise to find this
matrix. Just use your common sense and the definition of the R relation.

Part (b): Show the R matrix for the loop body — i.e., \( R_{Q_1=Q+1; R_i=R-N} \)
Again, just use your common sense and the definition of the R relation.

Part (c): In this part, you will use the matrix equations for a while loop
(slide 42) to illustrate how calculation of the R matrix for the while statement
in lines 6-9 works. Specifically, find the following matrices:

\[
\begin{align*}
(R_{Q_1=Q+1; R_i=R-N})' & \quad L_{R<=N} & \quad M_{R<=N} & \quad D_{Q_1=Q+1; R_i=R-N} \\
\end{align*}
\]

Then perform the matrix calculation: \((R_{Q_1=Q+1; R_i=R-N})'(L_{R<=N} M_{R<=N} D_{Q_1=Q+1; R_i=R-N} + I)\) to
find the \( R_{while..} \) for the while loop in line 6-9.

Part (d): Finally compute the product of the R matrices obtained in part (a) and
part (c) to find the R matrix for the procedure body.

What Depends aspect does this final matrix justify using for the Divide
procedure?