Cody Problem 34. Binary numbers

Given a positive, scalar integer n, create a (2^n)-by-n double-precision matrix containing the binary numbers from 0 through 2^n-1. Each row of the matrix represents one binary number. For example, if n = 3, then your code could return

>> binary_numbers(3)

ans =

```
%     1     1     1
%     0     0     0
%     0     1     1
%     0     1     0
%     0     0     1
%     1     0     0
%     1     1     0
%     1     0     1
```

Scratch Pad

```
n = 2;

A = binary_numbers(n)
```

```
A = 4×2
     0     0
     0     1
     1     0
     1     1
```

```
n = 3;

A = binary_numbers(n)
```

```
A = 8×3
     0     0     0
     0     0     1
     0     1     0
     0     1     1
     1     0     0
     1     0     1
     1     1     0
     1     1     1
```

```
n = 5;

A = binary_numbers(n)
```

```
A = 32×5
     0     0     0     0     0
```

```
0     0     0     0     1
0     0     0     1     0
0     0     0     1     1
0     0     1     0     0
0     0     1     0     1
0     0     1     1     0
0     0     1     1     1
0     1     0     0     0
0     1     0     0     1
⋮
```

Solution

```
function A = binary_numbers(n)
    lst = dec2bin(0:2^n-1);
    for i = 1:length(lst)
        for j = 1:length(lst(1,:))
            A(i,j) = str2num(lst(i,j));
        end
    end
end
```