Example 1

Write a program fragment that calculates the cumulative sums of a given vector $v$. The cumulative sums should be stored in a vector of the same length as $v$. E.g., the cumulative sums for the sequence 1,3,5,0 is 1,4,9,9. Do not use MATLAB predefined functions other than `length`.

Example 2

Write a function `evalPoly` to evaluate an $n$th order polynomial of $x$:

$$a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$

The input parameters are `coef` and `x` where `coef` has length $n+1$ and contains the coefficients of the polynomial and `x` is the value at which to evaluate the polynomial. Return the evaluated value. Note that `coef(1)` is the coefficient for the term $x^0$. Do not use MATLAB predefined functions other than `length`.

```matlab
function val = evalPoly(coef,x)
    % Post: val is polynomial of x given coefficients stored in coef
    % coef(1) + coef(2)*x + coef(3)*x^2 + ...
    % Pre: length(coef)>=1
```
A random walk with graphics

Write a function `randomWalk` that performs $n$ steps of a “random walk” starting from position $(x_0, y_0)$ and draws the path. In a random walk, possible moves are left, right, up, or down (in a Cartesian plane).

```matlab
function randomWalk(n,x0,y0)
    % Post: make n steps of random walk starting from position (x0,y0), show path
    % Pre: n>0
    % possible movements: ( deltaX(i), deltaY(i) )

deltaX=

deltaY=

    x= [x0 zeros(1,n)];  % trajectory in x direction
    y= [y0 zeros(1,n)];  % trajectory in y direction

    % Perform walk, each step is based on a random integer
    for k = 2:n+1
        % get a random integer in (1..4)
        r=

        % take the step
        x(k)=

        y(k)=
    end

    % Show the walk
    plot(x,y,x(1),y(1),'r*',x(end),y(end),'ro')
    axis('equal')
    title([num2str(n) ' steps of random walk from * to o'])
```