Exercise 4 - Implementing a Random Walk

To visualise a random walk, you could use the following algorithm:

- Set `nsteps` to some integer value (e.g, 20)
- Set `x` to a an arbitratry vector of "length" `nsteps` – it could be filled with zeros
- Set `pos`, which keeps track of the current position to 0

```
for i from 1 to nsteps

    r = one_or_two % randomly chosen

    if r is 1 then
        pos = pos + 1
    if r is 2 then
        pos = pos - 1

    x(i) = pos

plot(x)
```

Hints and Useful Functions

- Use `randi(2)` to simulate the flipping of a coin. It returns either 1 or 2!
- You may use the += and `-=` operators. `x += 1` is equivalent to `x = x + 1`

```
nwalks = 100;
hold on
for i = 1:nwalks


% ------------- YOUR CODE HERE -------------- %

    nsteps = 100;
    pos = 0;
    x   = zeros(1, nsteps);
    % Loop goes after this line









    % Loop ends before this line
    plot(x)

% ---------------------------------------- %



end
print -dpng 'fig.png'
```