count

syntax: `count(vector)`

purpose: This program simply counts the number of elements in `vector` whose value equals a particular number or falls into a specified range. To be more precise, `count` returns the number of elements whose value is non-zero, but this can be used with other programs to a great variety of purposes.

examples: ≫ `x = [1 2 3 4 5 6 7 20];`
≫ `count(x>5)` ⇒ *ans: 3*
≫ `count(x>=5)` ⇒ *ans: 4*
≫ `y = [1.1 2.1 3.1 4.0 4.9 5.8 6.7 19];`
To count how many elements of `x` are less than or equal to the corresponding elements of `y`:
≫ `count( x <= y )` ⇒ *ans: 4*

You have just lost $5 to a street gambler in a coin flip game. He tossed 5 heads in a row; the bet was that he wins $1 for each head while you win $1 for each tail. While walking home you realize that his coin was probably not fair. Putting aside for the moment the considerable circumstantial evidence that the gambler was a cheat, do you have any statistical justification for thinking that the coin was not fair?

```
nullhypoth = [0 1];  % equal probs of heads or tails
Ntrials = 1000;
z = starttally;
for trial=1:Ntrials
  flips = sample(5, nullhypoth);
  a = count(flips==0);  % count the number of heads
  tally a z;
end
count(z>=5)/Ntrials;  % p-value for 5 heads in 5 flips
```

The answer was `0.0310`, quite close to the widely known exact value of $\frac{1}{32}$ for the probability of getting 5 heads in 5 coin tosses.

see also: `length(x)` tells how many elements there are in a vector.

`size(x)` tells the number of rows and columns in a vector or matrix

`sum(x)` gives the sum of the values in a vector.