(1a) Code prints out 5 probabilities, each representing the probability of obtaining an odd number, calculated by runnning `t` iterations of a die roll

```
for j = [10,50,100,500,1000]
    disp(length(find(mod(mod(floor(rand(j,1)*1000),5)+1,2)==1))/j);
end
```

```
    0.4000

    0.6000

    0.5700

    0.5760

    0.6180
```

(1b) Through mathematical analysis, it appears that the supposed probability should be 0.600 because there are 3 odd numbers and 5 total numbers. 3/5 = 0.600.

(1c) Yes, as `t` gets bigger, the output of my program gets closer and closer to 0.6. 0.5980 is very close to 0.6.

(1d) In this case, the probabilities for each number 1, 2, 3, 4, 5 are, respectively, 2x, 2x, x, x, x.

Let the integer output of our rand() function mod 7 represent the possible outcomes of our die. (0, 2) -> 1; (1, 3) -> 2, (4) -> 3, (5) -> 4, (6) -> 5. Splitting it up like this makes it so that if the outcome mod 7 is even, then the die roll is odd and vice versa.

```
for j = [10, 50, 100, 500, 1000]
    disp(length(find(mod(mod(floor(rand(j,1)*1000),7),2)==0))/j);
end
```

```
    0.7000

    0.5200

    0.5100

    0.5900

    0.5720
```

Through mathematical analysis, it appears as if the probability the outcome is odd is 4/7 because (2x + x + x) / (2x + 2x + x + x + x) = 4/7.

Our output actually does show this value. 4/7 is approximately 0.5714, and our output is 0.5770 which is fairly close.