```
%3.B
N = 1000;

p_1 = [1/6 1/6 1/6 1/6 1/6 1/6]; % probability of each face
P_1 = [0 cumsum(p_1)] ;
roll_1 = zeros(N,1); % vector to hold results of each roll
% (the memory is "pre-allocated" in this
% way to speed up Matlab processing


for i=1:N % loop over number of rolls
    x=rand(1,1);
    for j=2:length(P_1), % determine result of each roll
        if( (P_1(j-1)<x) & (x<P_1(j)) )
            roll_1(i) = j-1;
            break
        end
    end
end

p_2 = [0.1 0.2 0.4 0.3]; % probability of each face for rigged dice
P_2 = [0 cumsum(p_2)] ;
roll_2 = zeros(N,1);

for i=1:N % loop over number of rolls
    x=rand(1,1);
    for j=2:length(P_2), % determine result of each roll
        if( (P_2(j-1)<x) & (x<P_2(j)) )
            roll_2(i) = j-1;
            break
        end
    end
end

Togheter_roll = roll_1+roll_2;
Togheter_chance_odd_or_even = zeros(1,2);%First value is even, second is odd.

for i=1:N
    if ~mod(Togheter_roll(i),2)
        Togheter_chance_odd_or_even(1) = Togheter_chance_odd_or_even(1)+1;
    else
        Togheter_chance_odd_or_even(2) = Togheter_chance_odd_or_even(2)+1;
    end
end

figure(1)

%3.D
bar(Togheter_chance_odd_or_even) %First value is even, second is odd.

Togheter_chance_odd_or_even_2 = zeros(1,2); %First value is greater than 2, second is 2.
```

```
for i=1:N
    if roll_2(i) > 2 && ~mod(Togheter_roll(i),2)
        Togheter_chance_odd_or_even_2(1) = Togheter_chance_odd_or_even_2(1)+1;
    else
        Togheter_chance_odd_or_even_2(2) = Togheter_chance_odd_or_even_2(2)+1;
    end
end

figure(2)
bar(Togheter_chance_odd_or_even_2) %First value is greater than 2 and even, second is↙
everything else.
```