%Assignment\_1 Problem A

clear all;close all;clc;%close all of windows

n=50;%toss the coin for 50 times

m=0;%the toss index is begin at 0 for calculate the HEAD times

%in this condition, assume the diameter of one coin is 1cm

for i=1:n;%begin to toss the coin

x=randperm(2)-1;

y=x(1);

if y==0;

m=m+1;

end

end

l\_head=1\*m;%calculate the total length of HEAD

fprintf('head shows: %f\n',m);

fprintf('the length of head in cm: %f\n',l\_head);

%%

%%problem A (solution 2)

input=50;% toss a coin for 50 times

r=rand(1,50);% produce 50 numbers randomly between 0 to 1

%which used for produce probability

x=0;

for i=1:input

if r(i)>0.5 ;%assume the randome number value is bigger than 0.5, then

%we think it got HEAD

x=x+1;%calculate the number of HEAD each time

end

end

l\_head=x\*1;

fprintf('head shows: %f\n',x);

fprintf('the length of head in cm: %f\n',l\_head);

%%

%%Problem B

clear all;close all;clc;

repition=1000;%in this experiment we try for 1000 times

input=50;%every coin toss for 50 times in each experiment

heads=zeros(1,repition)% total number of heads for each experiment

longests=zeros(1,repition)%longest length of coin per/ex

head\_1=zeros(1,repition)% toss number until 1st head/ex

head\_2=zeros(1,repition)% toss number until 2st head/ex

head\_3=zeros(1,repition)% toss number until 3st head/ex

head\_4=zeros(1,repition)% toss number until 4st head/ex

head\_2\_sequence=zeros(1,repition)% toss number until 2 head appear/ex

head\_3\_sequence=zeros(1,repition)% toss number until 3 head appear/ex

head\_4\_sequence=zeros(1,repition)% toss number until 2 head appear/ex

for i=1:repition% run 1000 times

head=0;% begin index/ex

longest=0;% the longest length of heads/ex

length=0;% current total length/ex

for j=1:input % 50 times/ex and (1000\*50)

head\_or\_back=rand();%randomly produce number between 0 to 1

if(head\_or\_back>0.5)% use >0.5 to replace the head

head=head+1;%accumulate heads

length=length+1;%accumulate length in one ex

if(length==2&&head\_2\_sequence(i)==0)% 2 heads in sequence

head\_2\_sequence(i)=j;%get 2 heads in per ex(50 toss), for per run in 1000 times

end

if(length==3&&head\_3\_sequence(i)==0)% 3 heads in sequence

head\_3\_sequence(i)=j;

end

if(length==4&&head\_4\_sequence(i)==0)% 4 heads in sequence

head\_4\_sequence(i)=j;

end

else

if(length>longest)% keep the longest update with length

longest=length;% give total length to longest, and make sure always longest

end

length=0;%start new coin toss

end

if (head==1)% got 1 head

head\_1(i)=j;

end

if (head==2)% got 2 head

head\_2(i)=j;

end

if (head==3)% got 3 head

head\_3(i)=j;

end

if (head==4)% got 4 head

head\_4(i)=j;

end

end

if(length>longest)% keep the longest update with length

longest=length;% this is the last length and end with longest (total length)

end

heads(i)=head; %total heads number for each run (should be 1000)

longests(i)=longest;%total heads length for each run (should be 1000)

end

%plot the frequency distrabution of 1000 runs/one run with 50 toss

figure(1)

hist(heads);

title('total heads of 1000 run');

figure(2)

hist(longests);

title('total length of 1000 run');

figure(3)

hist(head\_1);

title('the number of tosses until 1st head for 1000 run');

figure(4)

hist(head\_2);

title('the number of tosses until 2st head for 1000 run');

figure(5)

hist(head\_3);

title('the number of tosses until 3st head for 1000 run');

figure(6)

hist(head\_4);

title('the number of tosses until 4st head for 1000 run');

figure(7)

hist(head\_2\_sequence);

title('the number of tosses needed until there are 2 heads');

figure(8)

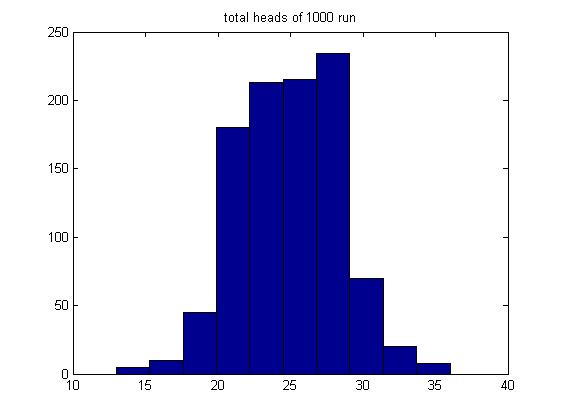
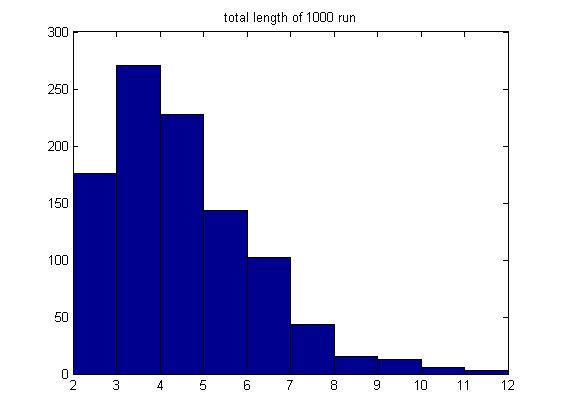
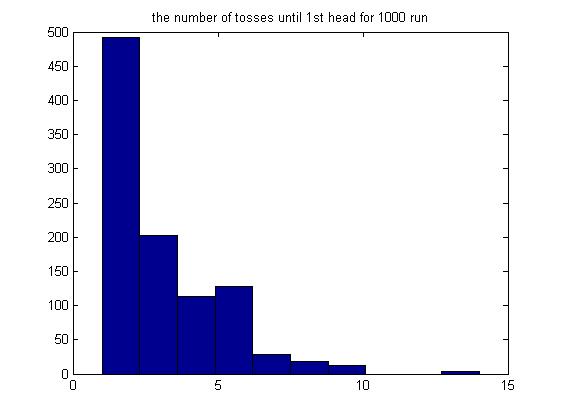
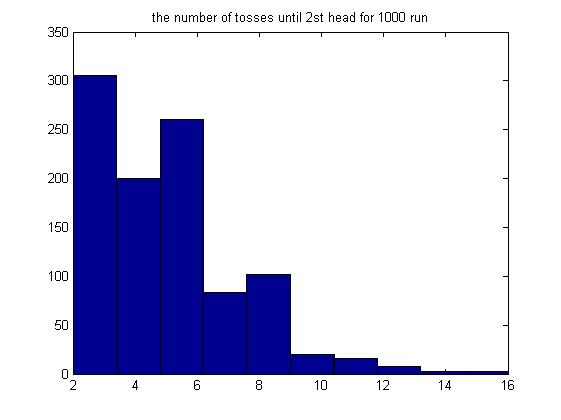
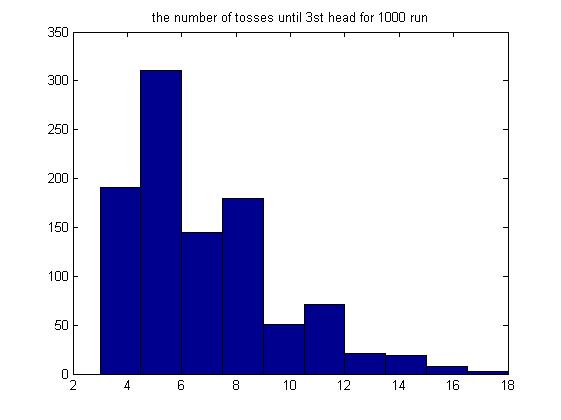
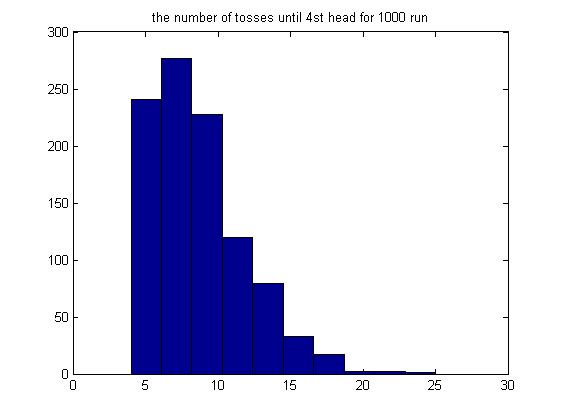
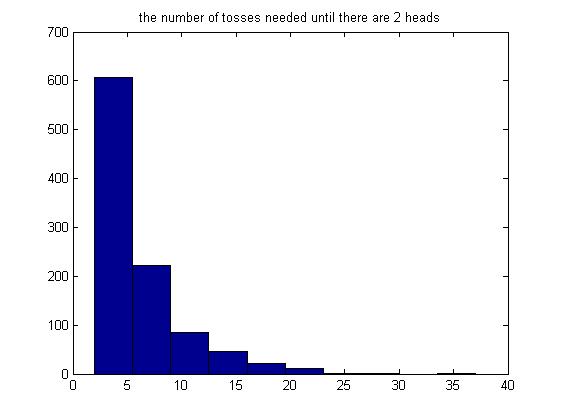
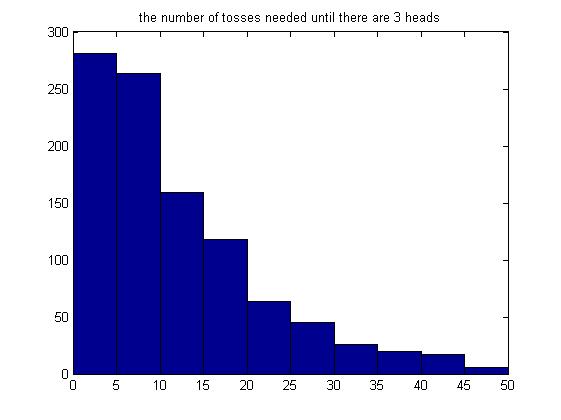
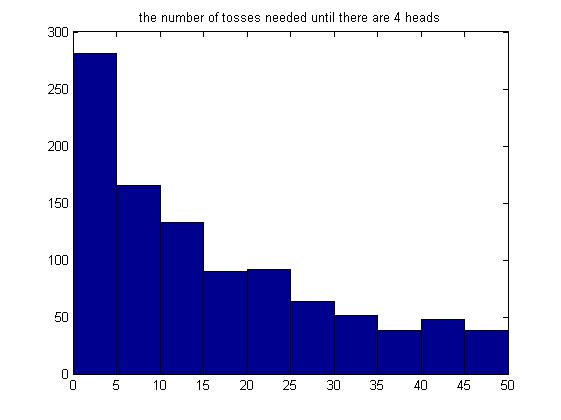
hist(head\_3\_sequence);

title('the number of tosses needed until there are 3 heads');

figure(9)

hist(head\_4\_sequence);

title('the number of tosses needed until there are 4 heads');

figure1figure 2figure3figure4figure5figure6figure7figure8figure9