Homework 2

Math 3607, Summer 2021

Spenser Smith

Table of Contents

Problem 1.

This problem asks for a script that simulates the tossing of a biased coin (where tails is 3x more likely than heads) and counts the number of each until a gap of 10 is reached.

```
tails=0;                        %initialize vars so that while loop functions properly
heads=0;
while abs(tails-heads) < 10     %while loop condition that checks if the gap is equal to 10
    x=rand;                     %generates random number b/w 0 and 1
    if x<=.75                   %since tails is 3x more likely its values are b/w 0 and 0.75
        tails=tails+1;
    else                        %heads gets the remaining possible values from 0.76 to 1
        heads=heads+1;
    end
end
fprintf('The number of tails is: %i \n', tails);
```

```
The number of tails is: 13
```

```
fprintf('The number of heads is: %i \n', heads);
```

```
The number of heads is: 3
```

Problem 2.

This problem asks for a script that prints the sequence until it is within a 1e-6 difference of π.

Part A uses the sequence: $a_n = \frac{6}{\sqrt{3}} \sum_{k=0}^{n} \frac{-1^k}{3^k(2k+1)}$

Part B uses the sequence: $b_n = 16 \sum_{k=0}^{n} \frac{-1^k}{5^{2k+1}(2k+1)} - 4 \sum_{k=0}^{n} \frac{-1^k}{239^{2k+1}(2k+1)}$.

```
%Part A
format long g;
k = 0;                      %initialize k to zero to get first value
array1=zeros(1,11);         %creates array to store data in easily
%finds first value so that while loop can function
array1(1) = (6/(sqrt(3)))*(((-1)^k)/((3^k)*(2*k+1)));
```

```
while((abs(array1(k+1)-pi)) > (10^-6))   %checks if condition is met
k = k + 1;                               %increments k
%adds the new value to the pre-existing one
array1(k+1) = array1(k) + (6/(sqrt(3)))*(((-1)^k)/((3^k)*(2*k+1)));
end
disp(array1);
```

```
  Columns 1 through 3

        3.46410161513775           3.079201435678          3.15618147156995

  Columns 4 through 6

        3.13785289159568          3.14260474566309          3.14130878546288

  Columns 7 through 9

        3.14167431269884          3.14156871594178          3.14159977381151

  Columns 10 through 11

        3.14159051093808          3.14159330450308
```

```
%Part B utilizes the same process at Part A just with a different sequence
k = 0;
array2=zeros(1,4);
array2(1) = ((16*((-1)^k))/((5^(2*k+1))*(2*k+1)))-...
            ((4*((-1)^k))/((239^(2*k+1))*(2*k+1)));

while((abs(array2(k+1)-pi)) > (10^-6))
k = k + 1;
array2(k+1) = array2(k)+(((16*((-1)^k))/((5^(2*k+1))*(2*k+1)))- ...
              ((4*((-1)^k))/((239^(2*k+1))*(2*k+1))));
end
disp(array2);
```

```
  Columns 1 through 3

        3.18326359832636          3.14059702932606          3.14162102932504

  Column 4

        3.14159177218218
```

Problem 3.

This problem asks for a script that find the probability that n number of people have different birthdays.

```
%Part A for n=30
prob=(365*364*363*362*361*360*359*358*357*356*355*354* ...
      353*352*351*350*349*348*347*346*345*344*343*342*341*340* ...
      339*338*337*336)/(365^30)
```

```
prob =
        0.293683757280731
```

```
%Part B for n=30
array3=zeros(1,30);          %creates array to store data
for k=1:30                   %runs loop 30 times since 30 birthdays
    x=randi(365);            %chooses random integer b/w 0 and 365
    array3(k)=x;             %stores the integer value inside the array
end
%checks if array length matches the length of unique numbers in array
if length(array3)==length(unique(array3))
   disp('There are no repitions')   %if yes, then displays there are no repetitions
else
%otherwise, finds the number of repititions and displays it
    y=length(array3)-length(unique(array3));
    fprintf('The number of repitions is: %i\n',y)
end
```

```
The number of repitions is: 1
```

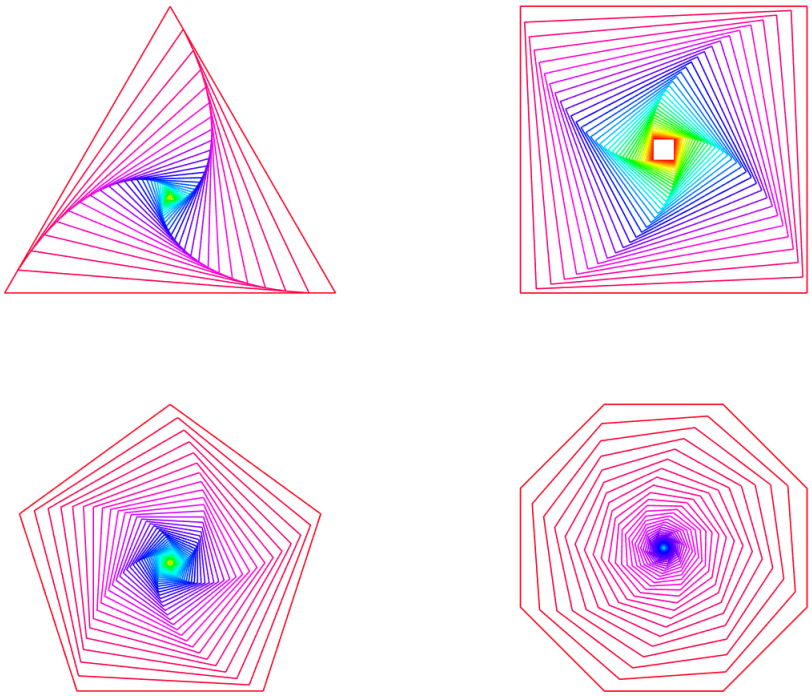
```
%Part C
numSims=input('Give me a number of simulations: ');   %asks for input
success=0;
failure=0;
for k=1:numSims                  %runs loop for however many times user wants
    array4=zeros(1,30);          %creates array to store data
    for n=1:30                   %same for loop as Part B
        x=randi(365);
        array4(n)=x;
    end

    if length(array4)==length(unique(array4))
        success=success+1;       %increments success if condition is met
    else
        failure=failure+1;
    end
end
probability=success/numSims;
fprintf('The probability of zero overlapping birthdays is %f\n',probability)
```

```
The probability of zero overlapping birthdays is 0.292940
```

Problem 4.

This problem asks for a script that generates spirals of n-gons for $n \geq 3$.

```
clf
subplot(2, 2, 1), spiralgon(3, 41, 4.5, -90);
subplot(2, 2, 2), spiralgon(4, 37, -2.5, 45);
subplot(2, 2, 3), spiralgon(5, 61, 3, -90);
subplot(2, 2, 4), spiralgon(8, 91, -4, 22.5);
```



```
function V = spiralgon(n, m, d_angle, d_rot)
% SPIRALGON plots spiraling regular n-gons
% inputs: n = the number of vertices
% m = the number of regular n-gons
% d_angle = the degree angle between successive n-gons
% (can be positive or negative)
% d_rot = the degree angle by which the innermost n-gon
% is rotated
% output: V = the vertices of the outermost n-gon

    th = linspace(0, 360, n+1) + d_rot; %n+1 so that spacing b/w vertices is correct
    V = [cosd(th);                      %everything else is the same as given
        sind(th)];
    C = colormap(hsv(m));
    s = sind(150 - abs(d_angle))/sind(30);
    R = [cosd(d_angle) -sind(d_angle);
        sind(d_angle) cosd(d_angle)];

    hold off
    for i = 1:m
        if i > 1
            V = s*R*V;
        end
        plot(V(1,:), V(2,:), 'Color', C(i,:))
        hold on
```

```
    end
    set(gcf, 'Color', 'w')
    axis equal, axis off
end
```