Question 1

```matlab
function [CorrectText]=CorrectText_(Text)
len=length(Text) %Length of text
for i=1:(len-1)
    cond1=(Text(i)=='.')|(Text(i)=='?')|(Text(i)==')')|(Text(i)==',') %Punctuation to look for
    cond2=(Text(i+1)~=' ') %Checking that there is no space after
    if cond1 & cond2 %if for punctuation sign followed by no space
        Text(i+2:end+1)=Text(i+1:end) %Shifting text up, and increasing length of text
        Text(i+1)=' ' % Setting position after punctuation equal to space
        len=length(Text) %Redefining length since Text is extended to add space
    end
end
first=1 %Idx for first position
len=length(Text) %Final length after all spaces have been added
if (Text(first)<'z') & (Text(first)>'a')
    Text(first)=Text(first)-'a'+'A' % Making sure first letter of script is a capital letter
end

for i=1:(len-2)
    cond3=(Text(i)=='.')|(Text(i)=='?')|(Text(i)==')') % Punctuations requiring capitalization
    cond4= (Text(i+2)<='z') & (Text(i+2)>='a') % Checking if lowercase after punctuation
    if (cond3 & cond4)
        Text(i+2)=Text(i+2)-'a'+'A' %Correcting from lowercase to capital
    end
end
CorrectText=Text %Defining output variable
```
% Any way to create these two vectors is fine

```matlab
clear; clc; format short;
n = 10
Enz = 'CCAA'
len = length(Enz);
a=.4;t=.1;c=.2;g=.3;
Seq = makeDNA_(a,t,c,g,n);
Size = 0; % same as doing Size=[]
for j = 1:length(Seq)-len+1
    if Seq(j:j+len-1) == Enz
        Size = [Size (j+1) - sum(Size)]; %since j is where the cut is, its not necessarily the length of the cut
    end
end
Size = [Size length(Seq) - sum(Size)]; %makes sure if the loop didnt work at the end it is included
Size(1) = []; %part of the same method of doing Size=[]
if Size(end) == 0
    Size(end) = [];
end
Seq
avg = mean(Size)
```
% Any way to create the vector is fine
Clear; clc;
Vec = input('Enter a string containing uppercase letters and digits');
kVal = 4;
[notEnough,startingAt] = ConsecutiveK_(Vec,kVal);

if notEnough
disp(['Less than ' num2str(kVal) ' consecutive letters or digits!']);
disp('The value of the second output has no meaning.');
else
disp(['Has at least ' num2str(kVal) ' consecutive letters or digits.']);
disp(['Starting at location: ' num2str(startingAt)]);
end

function [notEnough, atLocation] = ConsecutiveK_(Vec,minConsecutive)
lVec = length(Vec);
notEnough = true;
consecCounter = 1;
idx = 1
theElement = Vec(idx)

while notEnough && idx < lVec
    idx = idx + 1
    nextElement = Vec(idx)
    if nextElement - theElement == 1
        consecutiveCounter = consecutiveCounter + 1;
        notEnough = consecutiveCounter < minConsecutive;
    else
        consecutiveCounter = 1;
    end
    theElement = nextElement;
end

atLocation = idx - minConsecutive + 1;
return
Question 4

clear; clc; rng(5);

nJumps = 9;
maxValue = 6;
RESULTS = zeros(nJumps,3);

going2Right = true;

jumpBy = ceil(maxValue*rand);
xSquare = jumpBy;

RESULTS(1,:) = [1 jumpBy, xSquare];

for n = 2:nJumps
    newJump = ceil(maxValue*rand);
    if newJump > jumpBy % then reverse direction
        going2Right = ~going2Right;
    end
    
    if going2Right % Yes: moving the the right
        xSquare = xSquare + newJump;
    else % Moving to the left
        xSquare = xSquare - newJump;
    end
    % To get ready for the next move [6 pts]
    jumpBy = newJump;
    RESULTS(n,:) = [n jumpBy, xSquare];
end

disp(RESULTS);