



Polytechnic Tutoring Center

Final Exam REVIEW – CS1133, Spring 2020

Disclaimer: This mock exam is only for practice. It was made by tutors in the Polytechnic Tutoring Center and is not representative of the actual exam given by the Academic Department.

Question 1

NYU is tired of their students sending out ‘informal emails’ with their school address. Desperate to assure that the emails by students and faculty are at least properly spaced and capitalized, NYU superiors from Manhattan have asked you to develop a single function to correct a string text that has incorrect spacing and capitalization. For this particular question correct spacing is acknowledged as a space after any of these characters are used: ‘.’, ‘?’, ‘!’ and ‘,’. Then, correct capitalization is considered to be capitalization of the first letter after any of these characters are used: ‘.’, ‘?’ and ‘!’. You may not use vectorization.

Example output:

Initial Text:

hello Andrew.how have you been my good friend?congrats on your promotion.

Corrected Text:

Hello Andrew. How have you been my good friend? Congrats on your promotion.

Question 2

DNA is given by 4 letters (nucleotides) ATCG in a sequence. Write a function that generates a random sequence of given length with weights on each letter. Meaning we can have the sequence be about 40% A, 10% T, 20% C, and 30% G (in decimals so, .40, .10, etc.)

Restriction enzymes can cut a DNA sequence at specific locations. A random enzyme cuts DNA at –CCAA– resulting in the pieces –CC and AA–. Using the function above, run a trial where a DNA of length n is cut with a restriction enzyme and find the average length of the resulting fragments.

Have the weights of each letter be user inputted and have a data validation that makes sure the percentages total to 100.

Below is an example:

$n = 24$

Seq = CTGGACCAATTGACCATGCCAAT

Fragments: CTGGACC AATTGACCATGCC AAT

Average length: 8

Question 3

Write a function that determines if a given password has at least k consecutive letters or digits in ascending order. The function should also identify the locations of the problematic segment. If the password has less than the maximum number of consecutive uppercase letters or digits, the returned logical value should be false, and the location is meaningless.

For the above password, and if the maximum allowed consecutive letters or digits in ascending order is three, then the function should set and return a logical variable having the value of true, and a variable for the location of the problematic segment having the value of 6.

Write a script program to input a string consisting only of uppercase or digits from the user. Data validation is not needed. Then with the use the above function, determine whether the password is acceptable in the sense described here.

Here are three examples of the displayed results:

```
Enter a string of uppercase letters and digits: XKLM8MWQ45678ADV
Has at least 4 consecutive letters or digits.
Starting at location: 9
```

```
Enter a string of uppercase letters and digits: X012TKLMWQ94ADV
Less than 4 consecutive letters or digits!
The value of the second output has no meaning.
```

```
Enter a string of uppercase letters and digits: X012TKLMWQ94ADV
Has at least 3 consecutive letters or digits.
Starting at location: 2
```

Question 4

A long line of squares lies on the x -axis. Starting at the square at 0, a jump is made to square r to the right. The value of r is obtained by rolling a die. Subsequent moves are made according to the following procedure:

Roll a die to obtain a random integer r from 1 to 6. If r is higher than the previous roll, then make a jump in the opposite direction from the previous jump by r squares. Make a total of n jumps (including the first one). Record all your results in an n by 3 matrix named RESULTS. Row k of the matrix stores the move number, the row of the die, and the position after the move, respectively.

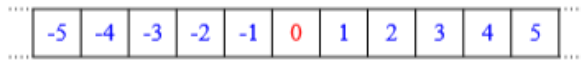


Figure 1: *A long line of squares on the x -axis. The starting position is on the red square at the origin.*

If you set the seed of the random number generator `rng(5)`, and for $n = 9$, you should get

1	2	2
2	6	-4
3	2	-6
4	6	0
5	3	3
6	4	-1
7	5	4
8	4	8
9	2	10