**Question:**

We use multiple parity bits and each parity bit is generated from a different subset of the original data bits. For example, consider the original data have 6 bits. We add two parity bits: The first one is the parity bit of 1st, 2nd, 3rd, and 4th bits; the second one is the parity bit of 1st, 2nd, 5th, and 6th bits. If the original data is 111111, then, the parity bits will be 00. This code can also be represented in a table below. "Y" in column i and row j indicates the ith bit is in the subset to compute the jth parity. (In reality, this is called linear block code.)

1 2 3 4 5 6

Parity 1 Y Y Y Y N N

Parity 2 Y Y N N Y Y

Still we send the original data bits and parity bits to the receiver. Each bit is independently flipped with probability p. We define

Event A: None of the bits are flipped;

Event B: Some of the bits are flipped, and this is detected by the advanced parity check.

Event C: Some of the bits are flipped, but this is not detected by the advanced parity check.

Let p = 0.1. Through Python simulation, compute the probability of events A, B, and C, i.e., pA, pB, and pC.

* Comment
* Flag Question
* Flag for Copyright

**Subject:**

* Python Programming
* Due in 3 hours 30 minutes

**Reminder:** Don't copy and paste from somewhere else. Write original answers and cite sources.

**Answer:**

**The output:**

Parity 1: Y Y Y Y N N

Parity 2: Y Y N N Y Y

No of the bits not flipped: 8

(a). Probability when none of the bits are flipped: 0.4304672100000001

(b). Probability when flipped and detected: 0.28476639499999995

(c). Probability when flipped and not detected: 0.28476639499999995

If the original data is 111111, then, the parity bits will be 00.

No of the bits not flipped: 0

(a). Probability when none of the bits are flipped: 1.0

(b). Probability when flipped and detected: 0.0

(c). Probability when flipped and not detected: 0.0

***PLEASE REFER TO THE EXPLANATION PART FOR THE BETTER UNDERSTANDING.***

**Explanation:**

**Logic:**

As noted, it is **0.1** prob for bits to get flipped.

Bits will not flip **(1-0.1)=0.9** probability

Any bit will not flip **(0,9)\*(0.9)\*(0,9)\*......=... (8 times)**

The probability of some bits being reversed= **1-0.43= 0.57**

Bits being reversed are equally likely to get detected and not detected.

**Approach:**

* Creating the 2D list for storing the parity table
* probability is calculated in the function name **prob\_cal()** method

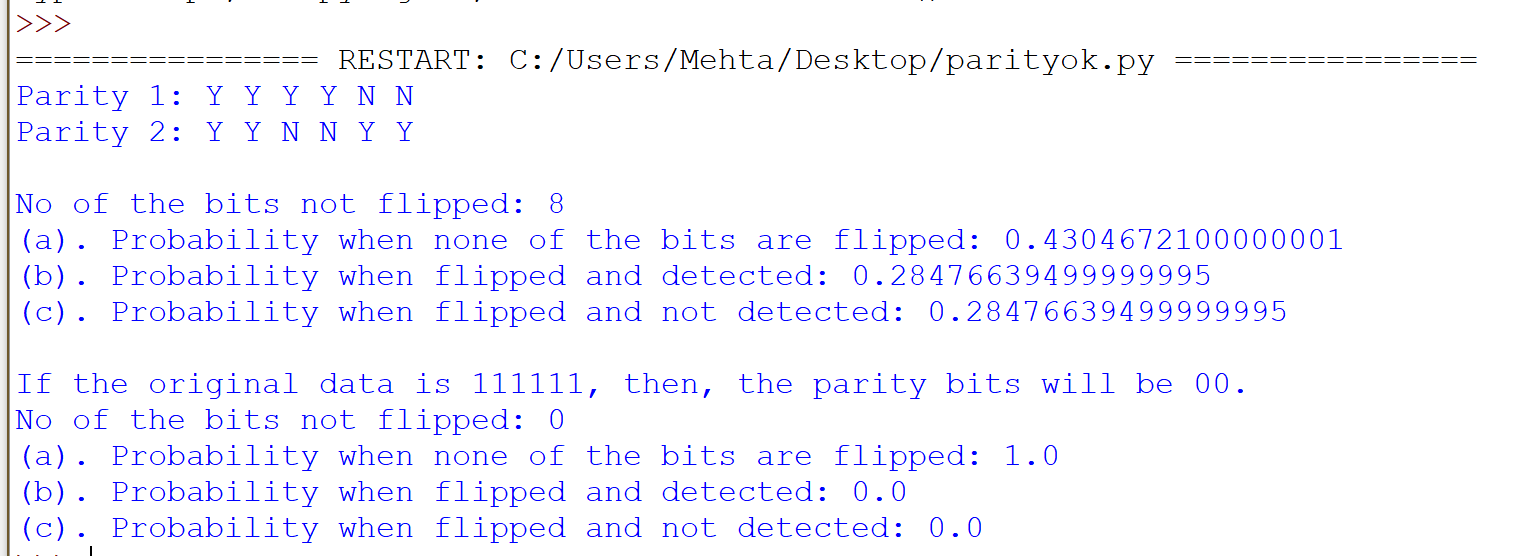
**NOTE: - count of Y is done by looping over the entire 2D list and then count the elements that have "Y" and then sum the count.**

**CODE:-**

#creating 2D list of the parity def linear\_block\_code(): #2D list parity = [["Y","Y","Y","Y","N","N"],["Y","Y","N","N","Y","Y"]] #printing the list #iterating over the row for i in range(len(parity)): #print the parity number as per the row number if ( i == 0 ): print("Parity 1: ", end ="") if ( i == 1): print("Parity 2: ", end ="") #iterating over the column for j in range(len(parity[i])): print(parity[i][j],end = ' ') print() #return the list return parity#probablity calculatordef prob\_cal(parity): #finding the number of Y in the list count = sum(elem.count('Y') for elem in parity) print("No of the bits not flipped:",count) #none flip prob given prob\_notflip = 0.1 #prob of event A pA = (1 - prob\_notflip)\*\*count print("(a). Probability when none of the bits are flipped:",pA) #prob of flip prob\_someflip = 1 - pA #"""There is fifty-fifty chance of getting detected by the parity check""" #prob of event B pB = prob\_someflip/2 print("(b). Probability when flipped and detected:",pB) #prob of event C pC = (1 - pA) \* (1/2) print("(c). Probability when flipped and not detected:",pC) #main methoddef main(): #calling the parity 2D list par = linear\_block\_code() print() #passing the par as parameter for calculating the parameter prob\_cal(par) print() ##checking the test case print("If the original data is 111111, then, the parity bits will be 00.") pari = [[1,1,1,1,1,1],[1,1,1,1,1,1]] prob\_cal(pari)

#running the main methodif \_\_name\_\_ == "\_\_main\_\_": main()

**OUTPUT: -**



* 999.PNG

[**Attach file(s)** or drag and drop file(s) in this box.](https://www.coursehero.com/qa/wait/18937094/)