

## # Algos-Task1-Questions

My solutions to the task 1 questions and an explanation following each of them and how they work. Explanations will be given in the readme file.

### #QUESTION 1

after taking the input, i use the following statement to check if the inputted string contains anything other than zero or one and then store those characters into a list k

```
statement: k=[i for i in a if (int(i)-int('0'))>1]
```

note: the logic behind is that when i force the character literal to be subjected to that convert to integer function in python, it will change it into its respective ascii value. Then i check if the difference between a given character's in the string's ascii value and the ascii value of 0 is more than 1 or not because this way every other character will give a value greater than 1.

then in the next line in the constraint checking/if loop starting statement i check if the length of the list is zero because if so then the string only contains '0's and '1's

```
statement: if len(k)==0 and a.count('0')!=n and a.count('1')!=n and n<=100000 :
```

note: this statement along with checking if the string contains only '1's and '0's also checks for other constraints given in the problem statement.

these next statements are because the strings don't follow the format in which binary is represented plus taking in the constraint of no leading zeroes we add "0b" to the string so that the int function can change the binary string to decimal

```
statements: b="0b"+a
```

```
p=str(bin(int(b,2)-1)).lstrip('0b')
```

```
q=str(bin(int(b,2)+1)).lstrip('0b')
```

note: +1 and -1 from the number because we are supposed to split given number into average of two different numbers, there can be many possibilities but we are supposed to output the ones with least difference between them which means it would have to be the number preceding and succeeding the given number.

the bin function is to then take the given number and turn it back to binary again and lstrip('0b') is to strip 0b from the string cuz of the output constraint and from observing the test case output.

```
statement: if len(p)!=len(q):
```

note: this is to check if the two output strings are of the same length or not.

### #QUESTION 2

we introduce a variable count of int type and initialize its value to zero and also another variable g whose value will be half of that of the inputted string's length.

in the if condition we check the constraints provided in the problem statement. (note: i made a mistake here i checked if n is a multiple of 2 instead of a power of 2, this can be done using the log

function , the statement instead should be  $(\log_{10}(n)/\log_{10}(2))\neq 0$  sadly i cant correct that mistake rn and thankfully there wasnt a no. like 18 in the test case)

for the  $2^0$  case where n is 1 and output will be 0 because i have intialized count to 0 and in this case it wont enter the loop in the first place the output comes out to be correct)

statement: for i in range (n//2):

```
b=a[:((g))]
```

```
c=a[-((g)):]
```

i run the for loop n/2 times because the value of the output (count) will always be less than n/2.

then i slice the string into two halves and put them into b and c respectively and then check if they are equal or not, if they are equal then the value of a changes to one of the equal halves and g gets halved again plus count increments. and if it isnt equal it will break from the loop and output the present value of count.

### #QUESTION 3

after taking the input , because the input is given in single line with each element seperated by a space i use the split function to split the input with respect to " " and store the individual elements in a seperate list j,k,l respectively

The following statement is to remove any extra spaces in the list that have entered because i noticed that there was one in the test case at the end of the input.:

if "" in k:

```
k.remove("")
```

if "" in l:

```
l.remove("")
```

because each element is still a character , the following statements are to use the map functions to operate the int function on the list and convert the elements into integer so that we can further check them for conditions and such .

the next two modules is to check and increment or decrement from the intial amount of money according to the conditions given in the problem statement and the second module of code is to check what should be outputed according to the increment or decrement or no change in the principle amount.

### #QUESTION 4

this code extremely brute force , i am going to be pointing at some redundant statements please excuse it as i wrote this code at 3 in the morning and i just now am realising the redundancy of these certain statements

the following statements are completely pointless and are from the earlier versions of the code that i wrote and were left in there :

if i<1:

```
print(s1+"*",s3,sep="")
```

```
if i<len(lst):
```

```
    print(s1,s2,s3,sep="")
```

after getting the number of test cases we store the inputs for the different test cases in the the list n through a for loop and then use a for loop again to send each test case to a function which will print the pattern for each of the test case.

in the funtion the first for loop snippet is to print the pattern row half+1 of the pattern and the second loop is to print the rest.

for eg: if n is 9 the first loop will print the pattern uptill row 5 and the second row will print uptill 4