18. You are given a linked list with head node as A, and also an integer B.

You need to change the value of each node to its nearest multiple of B that is ≥ current value.

ex: A: 1 → 2 → 3 / B: 2 / o/p → 2 → 2 → 4

A: 3 → 4 → 5 / B: 3 / o/p → 3 → 6 → 6.

idea 1:

Try incrementing every node value by 1 till it becomes a multiple of B.

A: 11 → 22 → 33 → 50

B: 10 .

idea 2:

[A[i] // B] → quotient.

ans = [quotient + 1] * B.

```
def modifyList ( A, B)
{
    current = A.
    while ( current != null)
    {
        if ( current.val % B != 0)
            quotient = (current.val // B)
            current.val = (quotient + 1) * B.
        current = current.next
    }
    return A
}
```

28. You are given a string 'A', how many contiguous character 'c' you can get by doing B number of changes in the string at the most.

Change: changing a character at any index in the given string.

ex:

A = OYOROOMS
B = 1
c = "O"

OYOROOMS

O

ans = 4.

A = "abacus"
B = 2
c = "a".

abacus

a a

ans = 4.

A = OYOROOMS
B = 2
c = "O"

OYOROOMS

ans = 6.

Ideal: Consider every substring, calculate otherChars in them,
If count(otherChars) ≤ B => it's a possible answer. Keep updating max^m.

OYOROOMS

B = 2.

how many other chars are present → OOMS. → 2

(Anything != 'O')

→ YOROOMS. → 4.

→ OYOROO → 2.

TC: $O(N^3)$.

⇓

Optimize a little $O(N^2)$.

```
def maxContinuousChars(A, B, C, n)
{
    ans = 0.
    for i in range(0, n)
        otherChars = 0.
        for j in range(i, n)
            // substring [i j].
            if(A[j] != c)
                others chars = others chars + 1
            if(otherschars ≤ B)
                ans = max(ans, j - i + 1)
            else
                break
    return ans;
}
```

OYOROOMS.

↓

c = 0.

(0)

OYOROOMS

2 otherchars.

TC: $O(N^2)$.

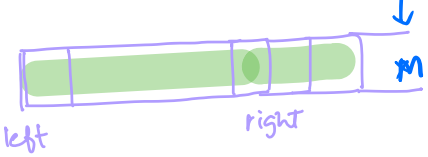
Can we improve?

O Y O R O O M S O O O O O O O S.

B = 2.

otherschars = 2.

M comes -> otherschars = 3.



(i) When A[right] == 'c'.

right ++

(ii) When A[right] != 'c'

otherChars ++

```
if (otherchars > B)
    // move left accordingly.

ans = max(ans, right - left + 1).
```

```
def maxContinuousChars(A, B, C)
{
    left = 0, right = 0, ans = 0, otherChars = 0.
    while(right < len(A))
    {
        if(A[right] != C)
            otherChars = otherChars + 1
        while (otherChars > B)
            if(A[left] != C)
                otherChars = otherChars - 1
            left = left + 1
        ans = max(ans, right - left + 1)
        right = right + 1
    }
    return ans
}
```

B = 2, C = 'O'

ans =

0	1	2	3	4	5	6	7	8	9	10	11
O	Y	O	R	O	M	O	O	O	O	O	$

l r

otherChars = 2

ans = 9