```
Slideng-Window ->X
            Input & Assay Siring
                                      output
                 number-k giver
                                            condition maxima
     fixeh Size window
                                                        minima
   ruryl sur mindow
                                                    ans=8713-20
                                         Sum = 6
                                                    1.1St windio0
              guz
                                                        Shrinle
               42
                            Sum=6+1-2=5
                                                                     2-2 20
                                =S+5 -3=7
                                                                     4-021
      784
                                =7+7-1213
                                                                       5-15
                                 - 13+8-1
                                   = 20+4-5517
= 20+4-5517
= 19+2-7=14
= 19+3-8=9
                                   sm=9 and =8123 13
public static void main(String[] args) {
     // TODO Auto-generated method stub
     int[] arr = { 2, 3, 1, 1, 5, 7, 8, 3, 4 };
     int k = 3;
     System.out.println(Maximum_Window(arr, k));
   }
  public static int Maximum_Window(int[] arr, int k) {
    \cap int sum = 0, ans = 0;
                                                          8+5-3=10+7-4
     for (int i = 0; i < k; i++) {
         sum += arr[i];
    Lans = sum;
     for (int i = k; i < arr.length; i++) {</pre>
         sum += arr[i];// grow
         sum -= arr[i - k];//. shrink
         ans = Math.max(ans, sum);
```

return ans;

Given an array of integers nums and an integer k, return the number of

```
contiguous subarrays where the product of all the elements in the
                                                                        int ans =0, P=1
subarray is strictly less than k.
1+2+3+1+2
                                                               while (cicarrilom) 2
                                                                   119204
                                                                     P=PXATICEI]
                                                                  11 Shoring
                                                                       P= 9/077(51)
                                                               nonsupdde winds ere
      3
                                                                 eith
```

```
public static int Product_Less_Than_K(int[] arr, int k) {
     int ans = 0, si = 0, ei = 0, p = 1;
     while (ei < arr.length) {</pre>
                                                    213
        // grow
        p = p * arr[ei]; \rightarrow 
        // shrink
           Si++;
        // ans update
        ans = ans + (ei - si + 1);
        ei++;
     return ans;
```

Kartik Bhaiya has a string consisting of only 'a' and 'b' as the characters. Kartik Bhaiya describes perfectness of a string as the maximum length substring of equal characters. Kartik Bhaiya is given a number **k** which denotes the maximum number of characters he can change. Find the maximum perfectness he can generate by changing no more than ${\bf k}$ characters.

```
0m2 7
      K=2 im si=01ei=0
                                                                     Simile Coics. 1000) &
                                                                        1/9820
if(8.chanAt(ei)==ch) L
                                                                      11 Shrine
While(flip>k) {
SiF(schnH(si)==ch)}
Fif(schnH(si)==ch)}
                                       abcaabede Ishaa
14
                                                                   is ansupate anscisin)
                                                                      eitt
3
```