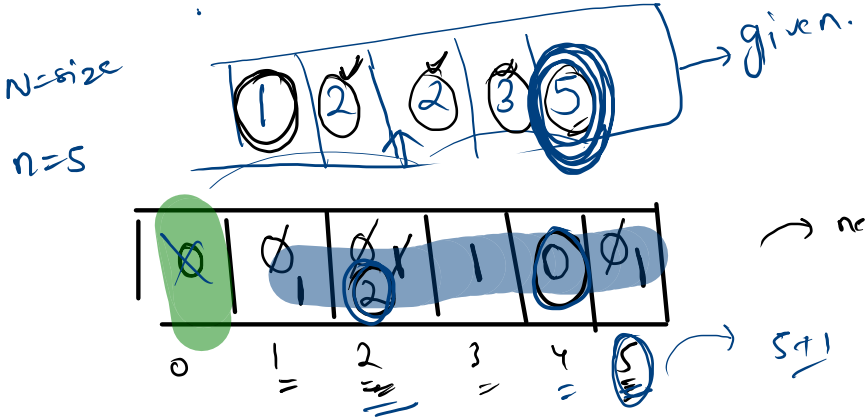
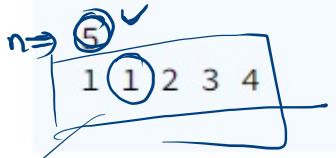
HW_Repeating and Missing element

Problem | Submissions | Leaderboard | Discussions

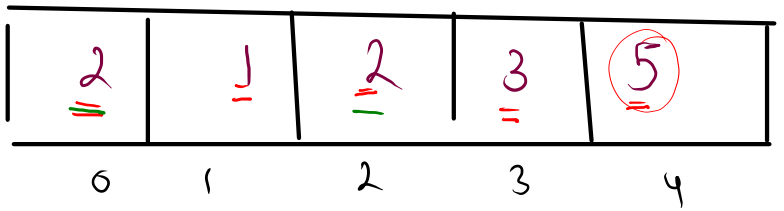
Given an array **arr[]** of size N of positive integers(1 - N). One number 'A' from set {1, 2, ...N} is **missing** and one number 'B' **occurs twice** in array. Write a program to print the **repeating element** and the **missing element** in arrav.



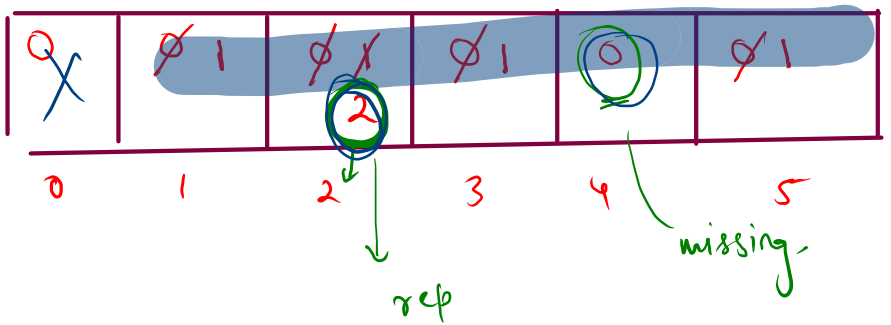


Sample Output 1

1
5



{1, N}

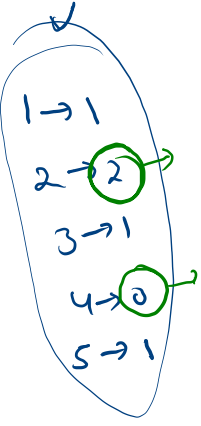


N = 5

5 1 3 3 2

? ? .

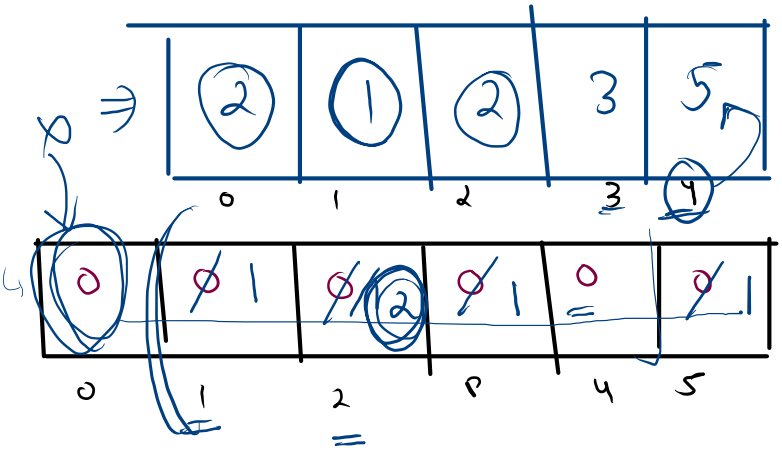
```
int [] freq = new int[n+1];

for(int i = 0; i < A.length; i++){
    int idx = A[i];
    freq[idx] = freq[idx] + 1;
}
```



i = 0 1 2 3 4

A[i] = 2 3 5

idx = 2 3 5



for

```
for(int i = 1; i < freq.length; i++){
    if(freq[i] == 0){
        missing = i;
    }
    else if(freq[i] == 2){
        repeating = i;
    }
}
```

age = 52

age = age + 1

$${}^nC_r = \frac{n!}{(n-r)!\,r!}$$

↳ ways to select r items from n items.

11th.

eg. $n = 4$

$r = 2$.

A B C D

AB AC AD | BC BO | CD

$${}^nC_r = \frac{n!}{(n-r)!\,r!}$$

$${}^4C_2 = \frac{4!}{2!\,2!}$$

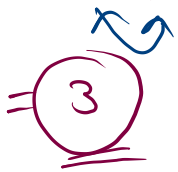
$$= \frac{4 \times (3 \times 2) \times 1}{2 \times 1 \times 2 \times 1}$$

^nC_r

eg. $n = 3$

$r = 2$

^nC_r $\qquad$ $^3C_2 = \dfrac{3!}{2!\ 1!}$

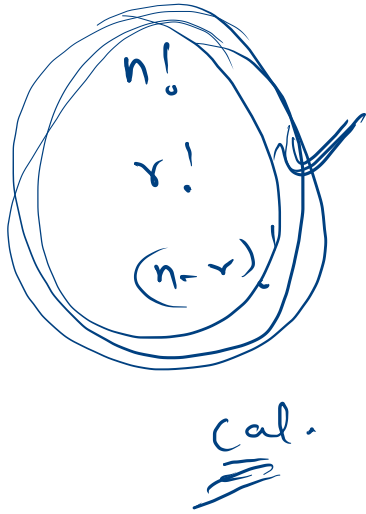


$\dfrac{3\cancel{!}\cancel{\times 2 \times 1}}{\cancel{2!} \times 1}$

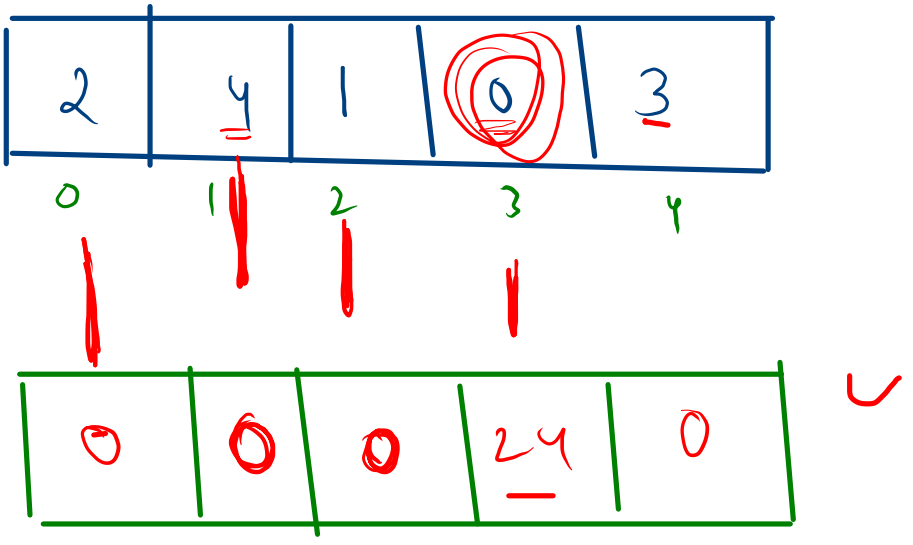
$$\frac{n!}{(n-r)!\,r!} = \frac{3!}{(3-2)!\ 2!}$$

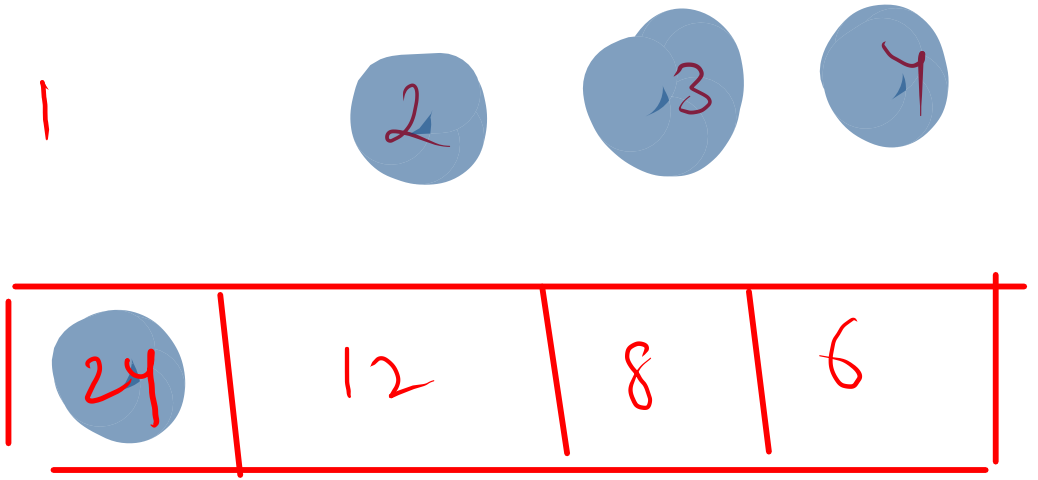
$$= \frac{3!}{1! \times 2!} = \frac{3!}{2! \times 1!}$$

```java
import java.io.*;
import java.util.*;

public class Solution {
    public static int factorial(int n){
        int prod = 1;
        for(int i = 1;  i <= n; i++){
            prod = prod * i;
        }
        return prod;

    }


    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int r = scn.nextInt();

        int nfact = factorial(n);
        int rfact = factorial(r);
        int nmrfact = factorial(n-r);

        int ans = nfact / (rfact * nmrfact);
        System.out.println(ans);

    }
}
```

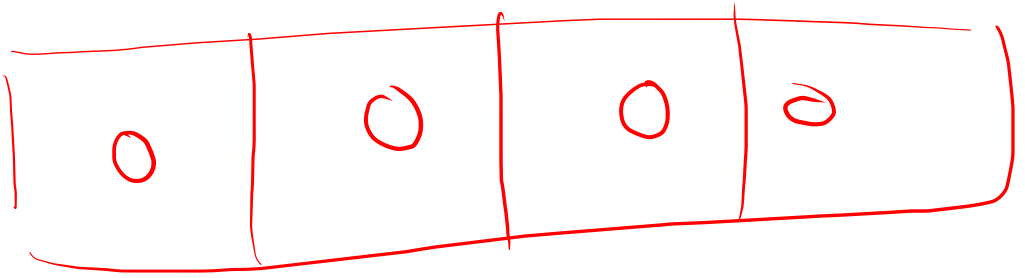


product except self




1
2
3
4
24
12
8
6

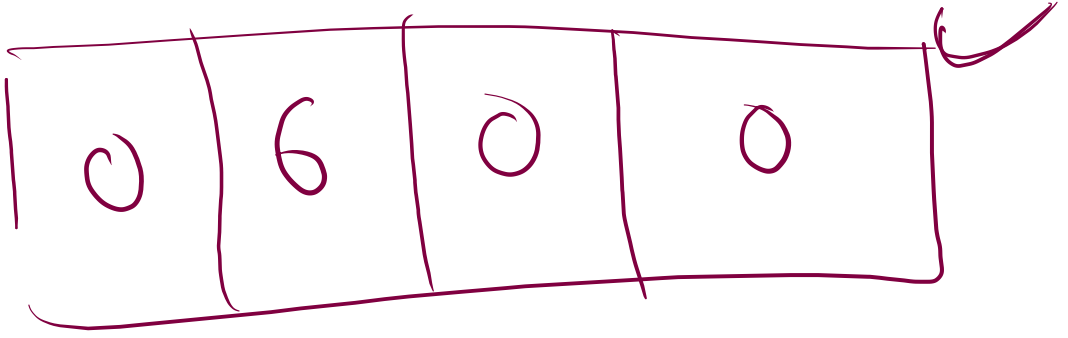
1 0 0 2

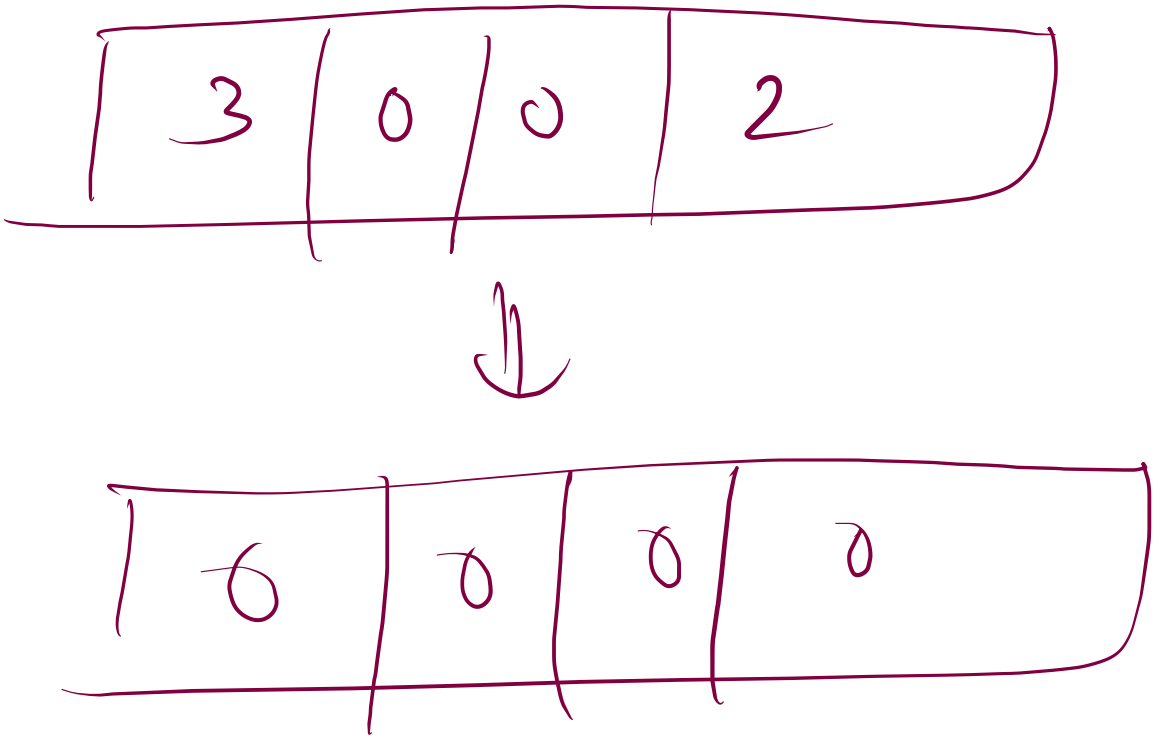


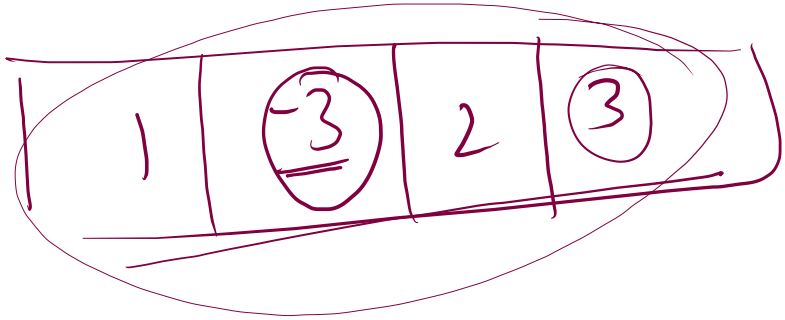
Your input

```
[-1,1,-3,3]
```


1
0
2
3


0
6
0
0


3
0
0
2
0
0
0
0



prod = -18

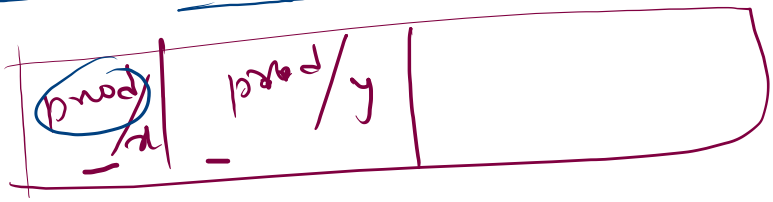
O(1).

-18/1 -18/-3 -18/2 -18/3

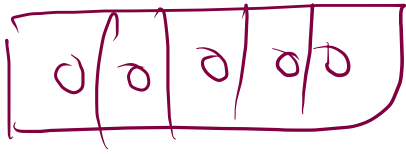
$$x * y = p$$

$$x = p / y.$$

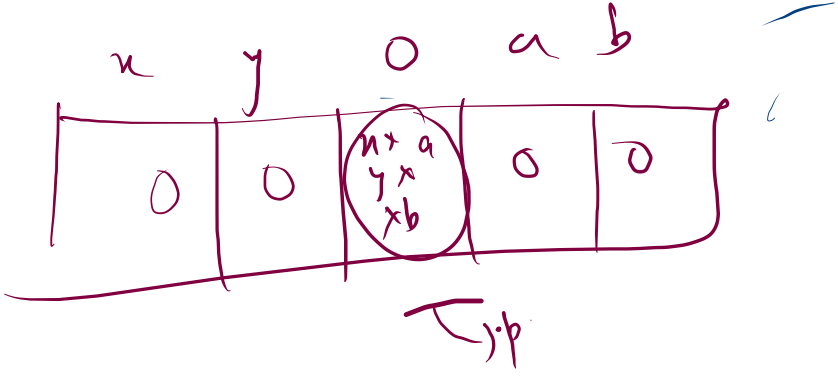
count zero = 0



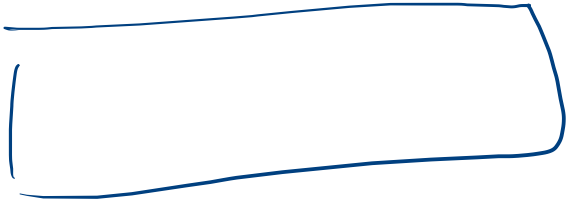
count zero = 2



count zero = 1

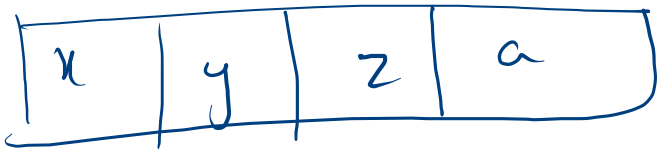


Typ



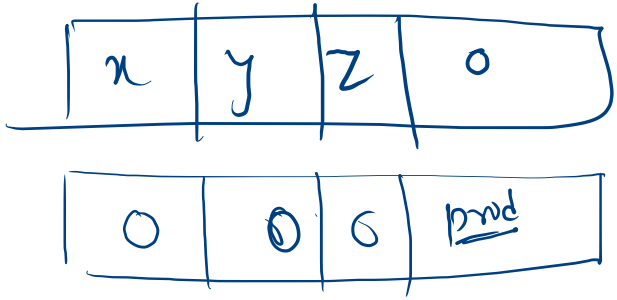
$$x * y * z = p$$

$$x = \frac{p}{y * z}$$



prod = x * y * a * 2

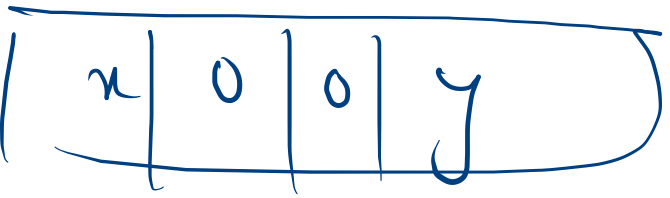
```
int count = 0;        //count of zero
int prod = 1;         // prod of non zero v

for(int i = 0; i < A.length; i++){
    if(A[i] == 0){
        count++;
    }
    else{
        prod *= A[i];
    }
}
```

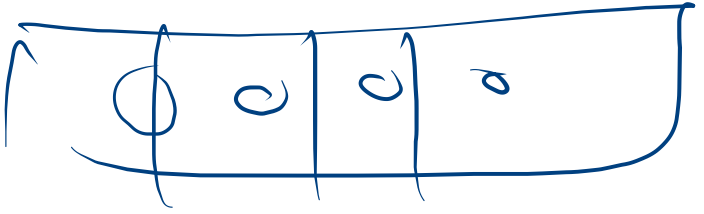


prod = x x y * 2

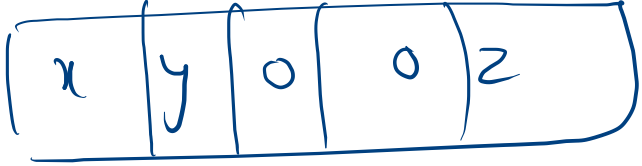
count = 1

```
int count = 0;        //count of zero
int prod = 1;         // prod of non zero v

for(int i = 0; i < A.length; i++){
    if(A[i] == 0){
        count++;
    }
    else{
        prod *= A[i];
    }
}
```


x
0
0
y

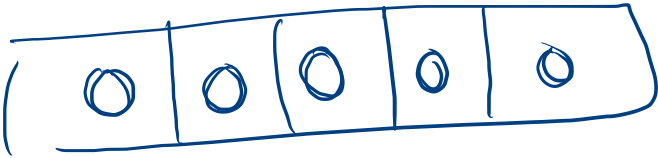


```
class Solution {
    public int[] productExceptSelf(int[] A) {

        int count = 0;      //count of zero
        int prod = 1;       // prod of non zero val

        for(int i = 0; i < A.length; i++){
            if(A[i] == 0){
                count++;
            }
            else{
                prod *= A[i];
            }
        }

        int [] ans = new int[A.length];

        if(count == 0){
            for(int i = 0; i < A.length; i++){
                ans[i] = prod / A[i];
            }
        }

        else if(count == 1){
            for(int i = 0; i < A.length; i++){
                if(A[i] == 0){
                    ans[i] = prod;
                }
            }
        }

        return ans;
```

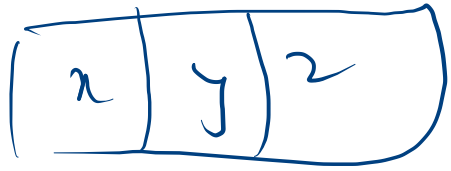


count = 2

prod = x * y * z (xyz)

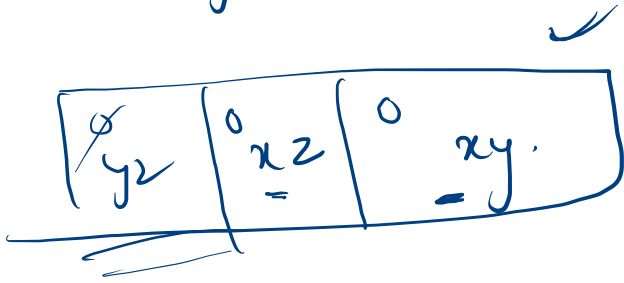


```
class Solution {
    public int[] productExceptSelf(int[] A) {

        int count = 0;      //count of zero
        int prod = 1;       // prod of non zero val

        for(int i = 0; i < A.length; i++){
            if(A[i] == 0){
                count++;
            }
            else{
                prod *= A[i];
            }
        }

        int [] ans = new int[A.length];

        if(count == 0){
            for(int i = 0; i < A.length; i++){
                ans[i] = prod / A[i];
            }
        }

        else if(count == 1){
            for(int i = 0; i < A.length; i++){
                if(A[i] == 0){
                    ans[i] = prod;
                }
            }
        }

        return ans;
```

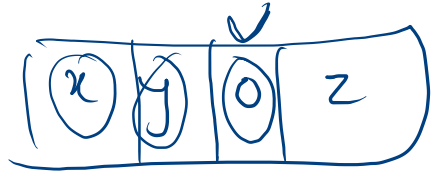


count = 0

prod = xyz



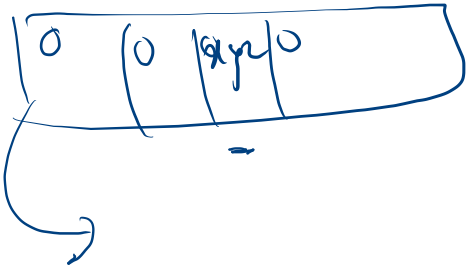
```
class Solution {
    public int[] productExceptSelf(int[] A) {

        int count = 0;      //count of zero
        int prod = 1;       // prod of non zero val

        for(int i = 0; i < A.length; i++){
            if(A[i] == 0){
                count++;
            }
            else{
                prod *= A[i];
            }
        }

        int [] ans = new int[A.length];

        if(count == 0){
            for(int i = 0; i < A.length; i++){
                ans[i] = prod / A[i];
            }
        }

        else if(count == 1){
            for(int i = 0; i < A.length; i++){
                if(A[i] == 0){
                    ans[i] = prod;
                }
            }
        }

        return ans;
```

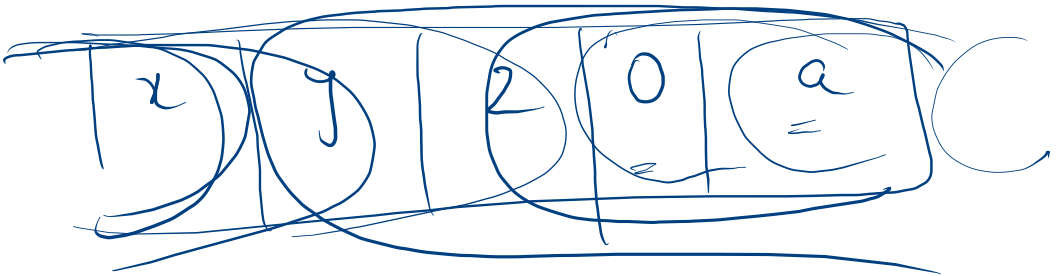


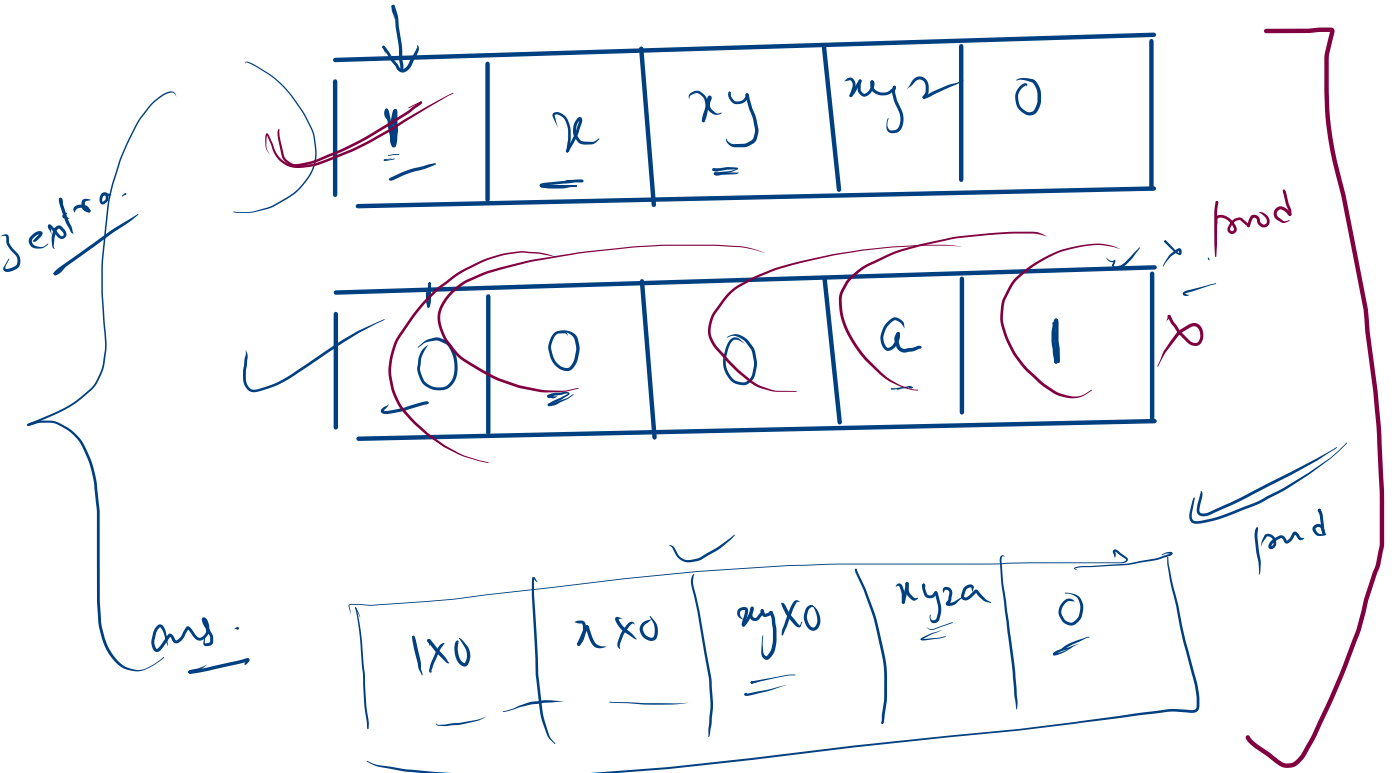
count = 1

A[i] == 0

prod = xyz




x
y
z
0
a


1
x
xy
xyz
0
0
0
0
a
1
found
ans.
1x0
xx0
xyx0
xyza
0
found