Revision

1. Rain water trapping
2. product of elements except itself.
3. Add one
4. Index as an element

Armstrong #

Ans = 0

(5, 6) - 5
0

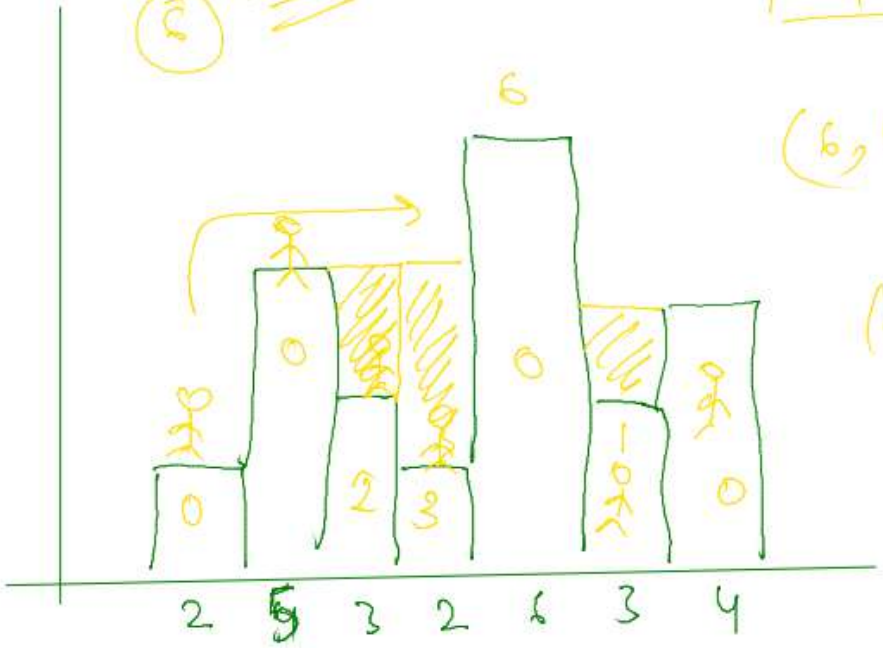
min(2, 6) - 2
2 - 2 = 0

(5, 6) - 3
5 - 3 = 2

(5, 6) - 2

5

Ans



| 2 | 5 | 3 | 2 | 6 | 3 | 4 |

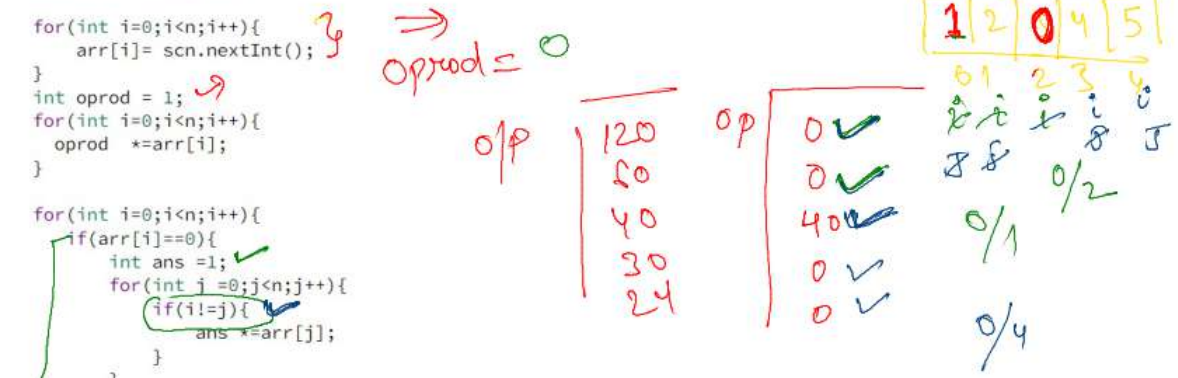
(6, 4) - 1)
1
(6, 4) - 4
= 0

Q 111

```
int[] arr = new int[n];

for(int i=0;i<n;i++){
    arr[i]= scn.nextInt();
}
int oprod = 1;
for(int i=0;i<n;i++){
  oprod  *=arr[i];
}

for(int i=0;i<n;i++){
   if(arr[i]==0){
       int ans =1;
       for(int j =0;j<n;j++){
           if(i!=j){
               ans *=arr[j];
           }
       }
       System.out.println(ans);
   }else{
       System.out.println(oprod/arr[i]);
   }
}
   /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your
}
```



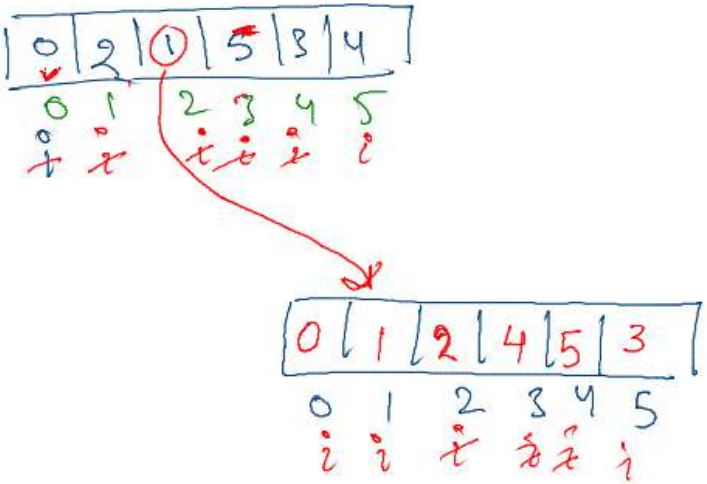
Qu Index as an element

```
6
0 2 1 5 3 4
0 1 2 3 4 5
```

num

Sample Output 0

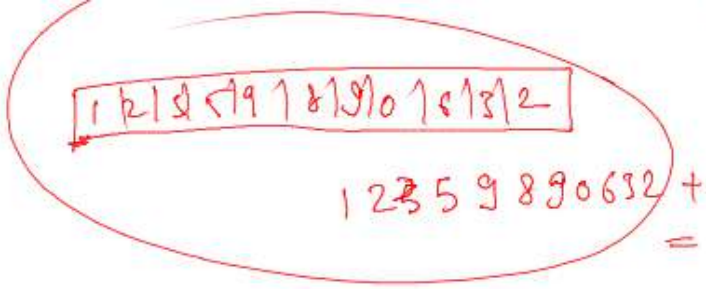
```
0 1 2 4 5 3
0 1 2 3 4 5
```




```
    int[] ans = new int[n];

    for(int i=0;i<n;i++){
        ans[i]=num[num[i]];
    }

    for(int i =0;i<n;i++){
        System.out.print(ans[i]+" ");
    }
    /* Enter your code here. Read input from STDIN. Print out
}
```



Qu Print all unique Prime factors.

i = 2 3 4 5 6 7 8

175 % 5 == 0

7 % 5 == 0

Prime factors

2	350
5	175
5	35
7	7
	1

```
while (n != 1) {
    if (n % i == 0) { Syso(i)
        while (n % i == 0)
            n = n / i;
    }
    i++;
}
```

2
5
5
7

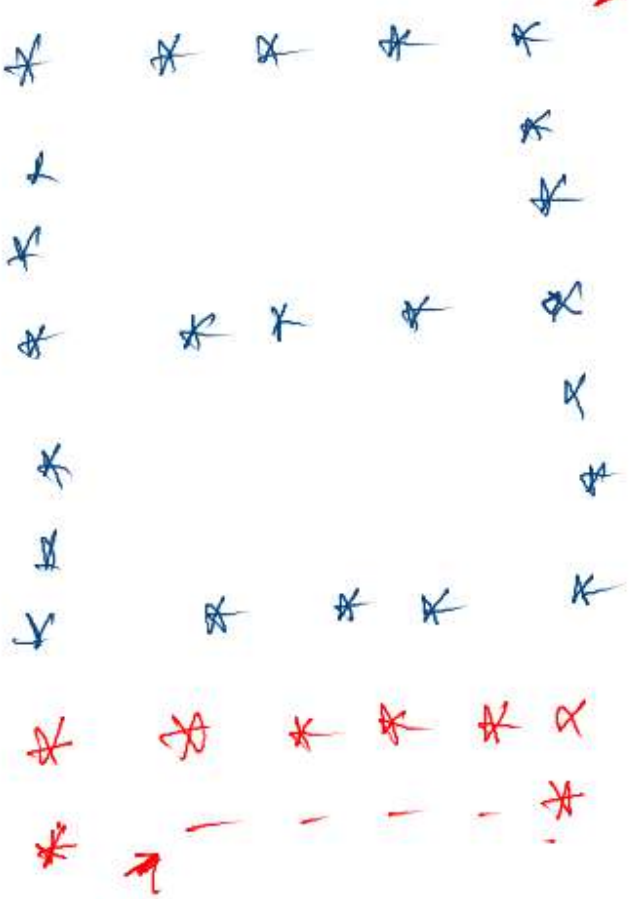
Qu

0	1	2	3	4	5	6	7
a	b	c	d	e	f	g	h

```
for( i=0 to arr.length-1) {
    int val = arr[i];
    for(2, val) {
        if( val % i == 0) {
            if( composite
                Syso ( val);
        }
    }
}
```

i = 1 2 3 ==1

j = 1 2



```
for(i=1 to n){
  for(j=1 to m){
    if(i%3==1 || j==1 || j==m){
      System.out.println("*");
    }
    else{
      System.out.println(" ");
    }
  }
  System.out.println();
}
```