Daily Practice Problem Date 9/12/2023

Problem 1: Minimum number of operations to make array equal to 1

```
package javaPractic;
import java.util.*;

public class MinOperation {    //class created

      public static int minOperation (int arr[], int n){    // function for min
operation

      HashMap<Integer, Integer> hash = new HashMap<Integer, Integer>();    // Insert all
elements in hash.

            for (int i=0; i<n; i++)     //loop iterating
                  if(hash.containsKey(arr[i]))
                        hash.put(arr[i], hash.get(arr[i])+1);
                  else hash.put(arr[i], 1);

            int max_count = 0;    // find the max frequency
            Set<Integer> s = hash.keySet();

            for (int i : s)
                  if (max_count < hash.get(i))max_count = hash.get(i);

            return (n - max_count);   // return result
      }


      public static void main(String[] args) {    //main program created
            int arr[] = {1, 5, 2, 1, 3, 2, 1};
            int n = arr.length;
            System.out.print(minOperation(arr, n));    //function calling

      }
}
```

Problem 2: Distinct prime factors of product of array

```
package javaPractic;
import java.util.*;

public class DistinctPrimeFactors {    //class created

      static int Distinct_Prime_factors(Vector<Integer> a){    //function created

            HashSet<Integer> m = new HashSet<Integer>();   // use set to store distinct
factors

            for (int i = 0; i < a.size(); i++) {       // iterate over every element of
array
                  int sq = (int)Math.sqrt(a.get(i));

                  for (int j = 2; j <= sq; j++) {    //loop iterating
                        if (a.get(i) % j == 0) {
                              m.add(j);

                              while (a.get(i) % j == 0) {
                                    a.set(i, a.get(i) / j);    //checking condition
                              }
                        }
                  }
                  if (a.get(i) > 1) {  //if true then execute below statements
                        m.add(a.get(i));
                  }
```

```java
                }

                return m.size();
        }

        public static void main(String args[]){      //main program created
                Vector<Integer> a = new Vector<Integer>();
                a.add(1);
                a.add(2);
                a.add(3);
                a.add(4);
                a.add(5);
                System.out.println(Distinct_Prime_factors(a));    //function calling
        }
}
```