1. Two Sum Problem
2. *public class* TwoSum {  
     
    *static class* Pair{  
    *int* a;  
    *int* b;  
     
    *public* Pair(*int* a, *int* b) {  
    *this*.a = a;  
    *this*.b = b;  
    }  
    }  
     
    *public static void* main(String[] args) {  
     
    System.***out***.println("Enter no of cases");  
    Scanner sc = *new* Scanner(System.***in***);  
    *int* no\_cases = sc.nextInt();  
    System.***out***.println("Enter size of array");  
    *int* size = sc.nextInt();  
    System.***out***.println("Enter target");  
    *int* target = sc.nextInt();  
    System.***out***.println("Enter array");  
    *int*[] my\_array = *new int*[size];  
    *for* (*int* i = 0; i < size; i++) {  
    *int* enter\_no = sc.nextInt();  
    my\_array[i] = enter\_no;  
    }  
    System.***out***.println(Arrays.*toString*(my\_array));  
    HashMap<Integer, Integer> hmap = *new* HashMap<Integer,Integer>();  
     
   *// two\_sum( my\_array,target,size );  
    for*(*int* i=0;i<size;i++){  
    *int* new\_b = target - my\_array[i];  
    *if*(hmap.containsValue(new\_b)){  
    System.***out***.println("("+new\_b+","+my\_array[i]+")");  
    }  
    *else*{  
    hmap.put(i,my\_array[i]);  
    }  
    }  
    }  
     
    *private static void* two\_sum(*int*[] my\_array, *int* target, *int* size) {  
    HashMap<Integer, Integer> hmap = *new* HashMap<Integer,Integer>();  
    ArrayList<Pair> my\_list = *new* ArrayList<Pair>();  
    *for*(*int* i=0;i<size;i++){  
    *int* new\_b = target - my\_array[i];  
    *if*(hmap.containsValue(new\_b)){  
    System.***out***.println("("+new\_b+","+my\_array[i]+")");  
    }  
    }  
     
    }  
   }

2.Maximum profit stock

*public class* MaxProfitStock{  
 *public static void* main(String[] args) {  
 Scanner sc = *new* Scanner(System.***in***);  
 System.***out***.println("Enter size");  
 *int* size = sc.nextInt();  
 System.***out***.println("Enter prices array");  
 *int*[] price\_arr = *new int*[size];  
 *for* (*int* i=0;i<size;i++){  
 price\_arr[i]= sc.nextInt();  
 }  
 System.***out***.println("Input: prices ="+Arrays.*toString*(price\_arr));  
 *int* min= Integer.***MAX\_VALUE***;  
 *int* max=Integer.***MIN\_VALUE***;  
 *int* max\_profit = 0;  
 *for*(*int* i=0;i<size;i++){  
  
 *if* (price\_arr[i] < min) {  
 min = price\_arr[i];  
 max= price\_arr[i];  
 }  
 *if* (price\_arr[i] > max) {  
 max = price\_arr[i];  
 }  
 *if*(price\_arr[i]>min){  
 max\_profit = max - min;  
 }  
  
 }  
 System.***out***.println(max\_profit);  
  
 }  
}

3.Merging two sorted array

4.Pascal triangle

*package* com.company.DSA;  
  
*import* java.util.ArrayList;  
*import* java.util.Arrays;  
*import* java.util.Scanner;  
  
*public class* Pascal{  
  
 *public static void* main(String[] args) {  
  
 Scanner sc = *new* Scanner(System.***in***);  
 *int* test\_case = sc.nextInt();  
 *int*[] test\_cases = *new int*[test\_case];  
 *for*(*int* i=0;i<test\_case;i++){  
 test\_cases[i]=sc.nextInt();  
 }  
  
 System.***out***.println(Arrays.*toString*(test\_cases));  
 *for*(*int* i=0;i<test\_case;i++){  
 *Print\_Pascal*(test\_cases[i]);  
 System.***out***.println();  
 }  
  
 }  
  
 *private static void* Print\_Pascal(*int* test\_case) {  
 ArrayList<Integer> arr\_list = *new* ArrayList<Integer>();  
 ArrayList<Integer> new\_list = *new* ArrayList<Integer>();  
 *for*(*int* i=0;i<test\_case;i++){  
 *if*(i==0){  
 System.***out***.println("1 ");  
 new\_list.add(0,1);  
 }*else if*(i==1){  
 System.***out***.println("1 "+"1 ");  
 new\_list.add(0,1);  
 new\_list.add(1,1);  
 }*else* {  
 *for* (*int* j = 0; j <= i; j++) {  
 *if*(j==0 || j==i){  
 System.***out***.print("1 ");  
 new\_list.add(j,1);  
 }*else* {  
 *int* new\_no =arr\_list.get(j)+arr\_list.get(j-1);  
 System.***out***.print(new\_no+" ");  
 new\_list.add(j,new\_no);  
 }  
   
 *if* (j==i){  
 System.***out***.println();  
 }  
  
 }  
 }  
  
 arr\_list.clear();  
 *for* (*int* k=0;k<new\_list.size();k++) {  
 arr\_list.add(k,new\_list.get(k));  
 }  
 new\_list.clear();  
 }  
 }  
}

5.Same concept as above stock

#### 6. You are given an array A of length N, where N is always an odd integer. There are (N-1)/2 elements in the array that occur twice and one element which occurs once. Your task is to find the only element that occurs once in the array.

*int*[] num = {7, 3, 5, 4, 5, 3, 4};  
*int* result =0;  
  
*for* (*int* ele: num  
 ) {  
 result = result^ele;  
}  
System.***out***.println(result);

7.Duplicate element

*int*[] num = {1,1,1,3,3,4,3,2,4,2};  
*int* result=0;  
HashMap<Integer,Integer> hmap = *new* HashMap();  
  
*for* (*int* i=0;i<4;i++) {  
 *if*(hmap.containsValue(num[i])){  
 System.***out***.println("true");  
 result=1;  
 *break*;  
 }  
 hmap.put(i,num[i]);  
}  
*if*(result!=1){  
 System.***out***.println("false");  
}

8.Majority element in array

*int*[] arr = {4,3,9,4,4};  
HashMap<Integer,Integer> hmap = *new* HashMap<>();  
  
*for*(*int* i=0;i<5;i++){  
 *if*(hmap.containsKey(arr[i])){  
 *int* value = hmap.get(arr[i])+1;  
 hmap.put(arr[i],value);  
 }*else* {  
 hmap.put(arr[i],1);  
 }  
}  
  
*for* (*Map*.*Entry* ele: hmap.entrySet()) {  
 *if*((*int*)ele.getValue()>1){  
 System.***out***.println((*int*)ele.getKey() );  
 }  
}

9.Duplicate element in array

Same as 7th question

10.Move 0 to left

*int*[] arr = {1,10,20,0,59,63,0,88,0};  
*int* count=arr.length-1;  
*for*(*int* i=arr.length-1;i>=0;i--){  
 *if*(arr[i]!=0){  
 arr[count--]=arr[i];  
 }  
}  
*while*(count>=0){  
 arr[count--]=0;  
}  
  
 System.***out***.println(Arrays.*toString*(arr));