1.import java.util.Arrays; /\*\* \* Java Program to find pairs on integer array whose sum is equal to given number

 public class ProblemInArray{ public static void main(String args[]) { int[] numbers = { 2, 4, 3, 5, 7, 8, 9 }; int[] numbersWithDuplicates = { 2, 4, 3, 5, 6, -2, 4, 7, 8, 9 }; prettyPrint(numbers, 7); prettyPrint(numbersWithDuplicates, 7); }

/\*\* \* Prints all pair of integer values from given array whose sum is is equal to given number. \* complexity of this solution is O(n^2) \*/

 public static void printPairs(int[] array, int sum) {

for (int i = 0; i < array.length; i++)

{ int first = array[i];

for (int j = i + 1; j < array.length; j++)

{ int second = array[j];

if ((first + second) == sum)

{ System.out.printf("(%d, %d) %n", first, second); } } } }

/\*\* \* Utility method to print input and output for better explanation.

 public static void prettyPrint(int[] givenArray, int givenSum){ System.out.println("Given array : " + Arrays.toString(givenArray)); System.out.println("Given sum : " + givenSum); System.out.println("Integer numbers, whose sum is equal to value : " + givenSum); printPairs(givenArray, givenSum); } }

2.find duplicate elements in array

public void findDupicateInArray(int[] a) {

//int pointer = a[0];

int count=0;

for(int j=0;j<a.length;j++) {

for(int k =j+1;k<a.length;k++) {

if(a[j]==a[k] && j!=k && j<k && count<=1) {

count++;

if(count==1)

System.out.println(a[j]);

}

}

\*\*count = 0;\*\*

}

}