Code : 1

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

public class CheckDuplicate {
   public static void main(String[] args){
      Scanner scn = new Scanner(System.in);
      int n = scn.nextInt();
      int[] arr = new int[n];
      for(int i = 0; i < arr.length; i++){
         arr[i] = scn.nextInt();
      }

      Arrays.sort(arr); // nlogn

      int dup = -1;
      for(int i = 0; i <= arr.length - 2; i++){ // n
         if(arr[i] == arr[i + 1]){
            dup = arr[i];
            break;
         }
      }

      System.out.println(dup);
   }
}
```

Code : 2

```
package Others;

import java.util.*;

public class CustomArrayList {
	public static class MyArrayList {
		int size;
		int[] data;

		static final int DEFAULT_CAPACITY = 5;

		MyArrayList() {
			this(DEFAULT_CAPACITY);
		}

		MyArrayList(int cap) {
			size = 0;
			data = new int[cap];
		}

		int get(int idx) {
			if (idx < 0 || idx >= size) {
				System.out.println("Invalid arguments");
				return -1;
			}
			return data[idx];
		}

		void set(int idx, int val) {
			if (idx < 0 || idx >= size) {
				System.out.println("Invalid arguments");
				return;
			}
			data[idx] = val;
		}

		int size() {
			return size;
		}

		void display() {
			for (int i = 0; i < size; i++) {
				System.out.print(data[i] + " ");
			}
			for (int i = size; i < data.length; i++) {
				System.out.print(" - ");
			}
			System.out.println();
		}

		void add(int idx, int val) {
			if (idx < 0 || idx > size) {
				System.out.println("Invalid arguments");
				return;
			}
```

```
                // if necessary resize
                if (size == data.length) {
                        System.out.println("Resizing up");
                        int[] ndata = new int[data.length * 2];
                        for (int i = 0; i < size; i++) {
                                ndata[i] = data[i];
                        }
                        data = ndata;
                }

                for (int i = size; i >= idx + 1; i--) {
                        data[i] = data[i - 1];
                }
                data[idx] = val;
                size++;
        }

        void remove(int idx) {
                if (idx < 0 || idx >= size) {
                        System.out.println("Invalid arguments");
                        return;
                }

                for (int i = idx; i <= size - 2; i++) {
                        data[i] = data[i + 1];
                }
                data[size - 1] = 0;
                size--;

                if (size == data.length / 4) {
                        System.out.println("Resizing down");
                        int[] ndata = new int[data.length / 2];
                        for (int i = 0; i < size; i++) {
                                ndata[i] = data[i];
                        }
                        data = ndata;
                }
        }
}

public static void main(String[] args) {
        // Write your code here
        MyArrayList list = new MyArrayList(4);
        list.add(0, 10);
        list.display();
        list.add(1, 20);
        list.display();
        list.add(2, 30);
        list.display();
        list.add(3, 40);
        list.display();
        list.add(4, 50);
        list.display();

        list.set(2, 300);
        list.display();
```

```
                list.add(2, 3000);
                list.display();

                list.add(4, 88);
                list.display();

                list.add(6, 34);
                list.display();

                list.add(2, 77);
                list.display();

                list.remove(1);
                list.display();

                list.remove(1);
                list.display();

                list.remove(1);
                list.display();

                list.remove(1);
                list.display();

                list.remove(1);
                list.display();

                list.remove(1);
                list.display();

                list.remove(1);
                list.display();

                list.add(1, 100);
                list.display();

                list.add(2, 200);
                list.display();

                list.add(0, 50);
                list.display();

        }
}
```

Code : 3

```
package Others;

import java.util.*;

public class FibonacciRecursion {
	public static void main(String[] args) {
		Scanner scn = new Scanner(System.in);
		int n = scn.nextInt();
		int fn = fib(n);
		System.out.println(fn);
	}

	public static int fib(int n) {
		if (n == 0 || n == 1) {
			return n;
		}

		int fnm1 = fib(n - 1);
		int fnm2 = fib(n - 2);
		int fn = fnm1 + fnm2;
		return fn;
	}
}
```

Code : 4

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

public class HFC {
   public static void main(String[] args){
      Scanner scn = new Scanner(System.in);
      String str = scn.next();

      int[] farr = new int[26]; // 0 pe a ki freq, 1 pe b ki freq
      for(int i = 0; i < str.length(); i++){
         char ch = str.charAt(i);
         int idx = ch - 'a';
         farr[idx]++;
      }

      int maxIdx = 0;
      for(int i = 1; i < farr.length; i++){
         if(farr[i] > farr[maxIdx]){
            maxIdx = i;
         }
      }

      char mfc = (char)(maxIdx + 'a');
      System.out.println(mfc);
   }
}
```

Code : 5

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

public class pmpwithjumps {

   public static void main(String[] args) throws Exception {
      Scanner scn = new Scanner(System.in);
      int row = scn.nextInt();
      int col = scn.nextInt();
                        printMazePaths(1, 1, row, col, "");
   }

   // sr - source row
   // sc - source column
   // dr - destination row
   // dc - destination column
   public static void printMazePaths(int sr, int sc, int dr, int dc, String psf) {
      if(sr == dr && sc == dc){
         System.out.println(psf);
         return;
      }

      for(int hss = 1; hss <= dc - sc; hss++){
         printMazePaths(sr, sc + hss, dr, dc, psf + "h" + hss);
      }

      for(int vss = 1; vss <= dr - sr; vss++){
         printMazePaths(sr + vss, sc, dr, dc, psf + "v" + vss);
      }

      for(int dss = 1; dss <= dr - sr && dss <= dc - sc; dss++){
         printMazePaths(sr + dss, sc + dss, dr, dc, psf + "d" + dss);
      }
   }

}
```

Code : 6

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


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

      int c = n;
      int pox = x;
      int ans = 0;
      while(c >= 1){
         int term = c * pox;
         ans += term;
         c--;
         pox = pox * x;
      }

      System.out.println(ans);
   }
}
```

Code : 7

```
package Others;

import java.util.*;

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

		int xpn = power3(x, n);
		System.out.println(xpn);
	}

	public static int power1(int x, int n) {
		if (n == 0) {
			return 1;
		}

		int xpnm1 = power1(x, n - 1);
		int xpn = xpnm1 * x;

		return xpn;
	}

	public static int power2(int x, int n) {
		if (n == 0) {
			return 1;
		}

		int xpb2 = power1(x, n / 2);
		int xpn = xpb2 * xpb2;

		if (n % 2 == 1) {
			xpn = xpn * x;
		}

		return xpn;
	}

	public static int power3(int x, int n) {
		if (n == 0) {
			return 1;
		}

		if (n % 2 == 0) {
			return power3(x, n / 2) * power3(x, n / 2);
		} else {
			return x * power3(x, n / 2) * power3(x, n / 2);
		}
	}
}
```

Code : 8

```
package Others;

import java.util.*;

public class Sortlohi {

	public static void sort012(int[] arr, int lo, int hi) {
		//write your code here
		int i = 0;
		int j = 0;
		int k = arr.length - 1;

		// 0 to j - 1 => is all 0's
		// j to i - 1 => is all 1's
		// i to k => unknowns
		// k + 1 to end => is all 2's
		while (i <= k) {
			if (arr[i] >= lo && arr[i] <= hi) {
				i++;
			} else if (arr[i] > hi) {
				swap(arr, i, k);
				k--;
			} else {
				// i.e it is 0
				swap(arr, i, j);
				i++;
				j++;
			}
		}

	}

	// used for swapping ith and jth elements of array
	public static void swap(int[] arr, int i, int j) {
		System.out.println("Swapping index " + i + " and index " + j);
		int temp = arr[i];
		arr[i] = arr[j];
		arr[j] = temp;
	}

	public static void print(int[] arr) {
		for (int i = 0; i < arr.length; i++) {
			System.out.println(arr[i]);
		}
	}

	public static void main(String[] args) throws Exception {
		Scanner scn = new Scanner(System.in);
		int n = scn.nextInt();
		int[] arr = new int[n];
		for (int i = 0; i < n; i++) {
			arr[i] = scn.nextInt();
		}
		sort012(arr, n, n);
		print(arr);
```

```
        }

}
```

Code : 9

```
package Others;

import java.util.*;

public class TargetSumPair {
	public static void main(String[] args) {
		Scanner scn = new Scanner(System.in);
		int tar = scn.nextInt();
		int n = scn.nextInt();
		int[] arr = new int[n];
		for (int i = 0; i < arr.length; i++) {
			arr[i] = scn.nextInt();
		}

		// sort
		Arrays.sort(arr); //nlogn

		// meet in the middle
		int left = 0;
		int right = arr.length - 1;
		while (left < right) { // n
			if (arr[left] + arr[right] > tar) {
				right--;
			} else if (arr[left] + arr[right] < tar) {
				left++;
			} else {
				System.out.println(arr[left] + " " + arr[right]);
				left++;
				right--;
			}
		}
	}
}
```

Code : 10

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


public class TargetSumPairWithBinarySearch {
   public static void main(String[] args){
      Scanner scn = new Scanner(System.in);
      int tar = scn.nextInt();
      int n = scn.nextInt();
      int[] arr = new int[n];
      for(int i = 0; i < arr.length; i++){
         arr[i] = scn.nextInt();
      }

      // sort
      Arrays.sort(arr); //nlogn

      // binary search the compliment
      for(int i = 0; i < arr.length; i++){
         int theOtherNumber = tar - arr[i];

         // binary search the new Target
         int left = i + 1;
         int right = arr.length - 1;
         while(left <= right){
            int mid = (left + right) / 2;
            if(theOtherNumber < arr[mid]){
               right = mid - 1;
            } else if (theOtherNumber > arr[mid]){
               left = mid + 1;
            } else {
               System.out.println(arr[i] + " " + arr[mid]);
               break;
            }
         }
      }

   }
}
```

Code : 11

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

public class TargetSumTriplet {
   public static void main(String[] args){
      Scanner scn = new Scanner(System.in);
      int tar = scn.nextInt();
      int n = scn.nextInt();
      int[] arr = new int[n];
      for(int i = 0; i < arr.length; i++){
         arr[i] = scn.nextInt();
      }

      // sort = nlogn
      Arrays.sort(arr);

      for(int i = 0; i < arr.length; i++){
         int ntar = tar - arr[i];

         int j = i + 1;
         int k = arr.length - 1;
         while(j < k){
            if(arr[j] + arr[k] > ntar){
               k--;
            } else if(arr[j] + arr[k] < ntar){
               j++;
            } else {
               System.out.println(arr[i] + " " + arr[j] + " " + arr[k]);
               j++;
               k--;
            }
         }
      }
   }
}
```