Code : 1

```
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 : 2

```
import java.io.*;
import java.util.*;

public class Partition {

	public static void partition(int[] arr, int pivot) {
		//write your code here
		int i = 0;
		int j = 0;
		while (i < arr.length) {
			if (arr[i] > pivot) {
				i++;
			} else {
				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 " + arr[i] + " and " + arr[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.print(arr[i] + " ");
		}
		System.out.println();
	}

	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();
		}
		int pivot = scn.nextInt();
		partition(arr, pivot);
		print(arr);
	}

}
```

Code : 3

```
import java.io.*;
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 : 4

```
import java.util.*;

public class power1 {
	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 : 5

```
import java.util.*;

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

		pss(str, "");
	}

	// xyz, .
	public static void pss(String ques, String asf, ArrayList<String> acont) {
		if (ques.length() == 0) {
			System.out.println(asf);
			return;
		}

		char ch = ques.charAt(0); // x
		String roq = ques.substring(1); // yz

		pss(roq, asf + ch);
		pss(roq, asf + "-");
	}

}
```

Code : 6

```
import java.util.*;

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

		ArrayList<String> acont = new ArrayList<>();
		pss(str, "", acont);
		System.out.println(acont);
	}

	// xyz, .
	public static void pss(String ques, String asf, ArrayList<String> acont) {
		if (ques.length() == 0) {
			acont.add(asf);
			return;
		}

		char ch = ques.charAt(0); // x
		String roq = ques.substring(1); // yz

		pss(roq, asf + ch, acont);
		pss(roq, asf + "-", acont);
	}

}
```

Code : 7

```
import java.io.*;
import java.util.*;

public class Sort01 {

	public static void sort01(int[] arr) {
		//write your code here
		int i = 0;
		int j = 0;
		while (i < arr.length) {
			if (arr[i] == 1) {
				i++;
			} else {
				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();
		}
		sort01(arr);
		print(arr);
	}

}
```

Code : 8

```
import java.io.*;
import java.util.*;

public class Sort012 {

 public static void sort012(int[] arr){
  //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] == 1){
        i++;
     } else if(arr[i] == 2){
        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);
  print(arr);
 }
```

```
}
```

Code : 9

```
import java.io.*;
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);
		print(arr);
	}
```

```
}
```