﻿

1.import java.util.Arrays;

import java.util.Scanner;

public class Lsolve1 {

public static void main(String[] args) {

int source [] = {10, 20, 30, 40, 50, 60};

source= shiftLeft(source,3);

System.out.println(Arrays.toString(source));

}

public static int [] shiftLeft (int source[], int a){

int i=1;

while (i <= a){

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

source[j-1] = source[j];

i++;

}

source[source.length-1] = 0;

}

return source;

}

}

2.import java.util.Arrays;

import java.util.Scanner;

public class Lsolve2 {

public static void main(String[] args) {

int source[] = {10,20,30,40,50,60};

source = rotateLeft(source,3);

System.out.println(Arrays.toString(source));

}

public static int[] rotateLeft(int[] s,int k){

for(int i = 0; i < k; i++){

int j, backup;

backup = s[0];

for(j = 0; j < s.length-1; j++){

s[j] = s[j+1];

}

s[j]=backup;

}

return s;

}

}

3.import java.util.Arrays;

import java.util.Scanner;

public class Lsolve3 {

public static void main(String[] args) {

int source [] = {10, 20, 30, 40, 50, 0, 0};

remove(source,5,2);

System.out.println(Arrays.toString(source));

}

public static int [] remove(int[] source,int size, int idx){

int i = idx;

while(i < source.length -1){

source[i]= source[i+1];

i++;

}

return source;

}

}

4.import java.util.Arrays;

import java.util.Scanner;

public class Lsolve4 {

public static void main(String[] args) {

int source [] = {10,2,30,2,50,2,2,60,0,0};

removeAll(source,8,2);

System.out.println(Arrays.toString(source));

}

public static int[] removeAll (int source[], int size, int value){

int[] notunArray = new int[source.length];

notunArray = source;

for(int t=0 ; t < notunArray.length ; t++){

if(notunArray[t] == value){

for(int j = t+1 ; j < notunArray.length ; j++){

notunArray[j-1] = notunArray[j];

}

t--;

}

}

return notunArray;

}

}

5.import java.util.Arrays;

import java.util.Scanner;

public class Lsolve5 {

public static void main(String[] args) {

int[] source ={10, 3, 10, 2, 1};

splittingArray(source);

}

public static void splittingArray(int[] s){

int lp\_sum = 0;

boolean sign = false;

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

{

lp\_sum += s[i];

int rp\_sum = 0;

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

rp\_sum += s[j];

}

if(lp\_sum == rp\_sum)

{

System.out.println(" ");

System.out.print("true");

sign = true;

break;

}

}

if(!sign) {

System.out.println(" ");

System.out.print("false");}

}

}

6.import java.util.Arrays;

import java.util.\*;

public class Lsolve6 {

public static void main(String[] args) {

Scanner sc =new Scanner(System.in);

System.out.println(" ");

int num = sc.nextInt();

arraySeries(num);

}

public static void arraySeries(int n){

int arr[] = new int[n\*n];

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

{

arr[i] = 0;

}

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

{

if((i+1)%n == 0)

{

int num = 1;

int backup = i;

for(int count = (i+1)/n; count > 0; count--)

{

arr[backup] = num++;

backup--;

}

}

}

System.out.println(" ");

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

{

System.out.print(arr[i]+" ");

}

}

}

7.import java.util.Arrays;

import java.util.Scanner;

public class Lsolve7 {

public static void main(String[] args) {

int[] source={1,1,2,2,1,1,1,1};

maxCounter(source);

}

public static void maxCounter(int[] s){

int counter = 0;

int maximum = 1;

for (int i = 0; i < s.length-1; i++) {

if (s[i] == s[i + 1]) {

counter++;

} else {

counter = 1;

}

if (counter > maximum){

maximum = counter;

}

}

System.out.println(" ");

System.out.println(counter);

}

}

8.public class Lsolve8

{

public static void main(String[] args)

{

int source[] = {4,5,6,6,4,3,6,4};

boolean result = repitition(source);

System.out.println(result);

}

public static boolean repitition(int source[])

{

boolean flag = true;

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

{

int count1 = 1;

int count2 = 1;

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

{

if(source[i]==source[j])

{

count1++;

//count2++;

}

count2++;

}

int count = 0;

if(count1==count2)

{

count=count1;

flag = true;

}

}

return flag;

}

}

Circular Array:

1.import java.util.Arrays;

import java.util.Scanner;

public class Csolve1 {

public static void main(String[] args) {

int[] num = new int[]{10,20,0,0,0,10,20,30};

boolean result = checkpalindrome(num,5, 5);

System.out.println(result);

}

public static boolean checkpalindrome(int[] num, int start, int length) {

boolean num1 = true;

int size = num.length;

int end = (start+length)-1;

for (int i = 0 ; i < (length/2); i = (i+1)%size){

if (num[i+start] != num[(end-(i))%size]){

num1=false;

return num1;

}

}

return num1;

}

}

2.public class Csolve2

{

public static void main(String[] args)

{

int array1[] = {40,50,0,0,0,10,20,30};

int array2[] = {10,20,5,0,0,0,0,0,5,40,15,25};

Intersection(array1,array2,5,5,8,7);

}

public static void Intersection(int[]a , int []b, int start1,int size1,int start2,int size2)

{

int index1 = start1;

for(int i=0;i<size1;i++)

{

int index2 = start2;

for (int j = 0; j < size2; j++)

{

if(a[index1]==b[index2])

{

System.out.println(a[index1]);

}

index2 = (index2 + 1)%b.length;

}

index1 = (index1 + 1)%a.length;

}

}

}