Министерство науки и высшего образования РФ

ФГАОУ ВО Пермский национальный исследовательский

политехнический университет

Кафедра «Вычислительная математика, механика и биомеханика»

Отчет по лабораторной работе № 8

тема «Одномерные массивы»

по дисциплине «Информатика»

Вариант 2

Выполнил: студент группу ИСТ-22-1б Афонина А.М

Проверил: доцент каф. ВММБ Каменских А.А

Пермь, 2023

**Содержание**

[Задание 1 3](#_Toc117979531)

[1.1. Постановка задачи 3](#_Toc117979532)

[1.2. Решение задачи, код программы 3](#_Toc117979533)

[1.3. Тестирование работы программы с проверкой 3](#_Toc117979534)

[Задание 2 5](#_Toc117979535)

[1.1. Постановка задачи 5](#_Toc117979536)

[1.2. Решение задачи, код программы 5](#_Toc117979537)

[1.3. Тестирование работы программы с проверкой 5](#_Toc117979538)

[Задание 3 7](#_Toc117979539)

[1.1. Постановка задачи 7](#_Toc117979540)

[1.2. Решение задачи, код программы 7](#_Toc117979541)

[1.3. Тестирование работы программы с проверкой 7](#_Toc117979542)

[Задание 4 9](#_Toc117979543)

[1.1. Постановка задачи 9](#_Toc117979544)

[1.2. Решение задачи, код программы 9](#_Toc117979545)

[1.3. Тестирование работы программы с проверкой 9](#_Toc117979546)

[Задание 5 11](#_Toc117979547)

[1.1. Постановка задачи 11](#_Toc117979548)

[1.2. Решение задачи, код программы 12](#_Toc117979549)

[1.3. Тестирование работы программы с проверкой 12](#_Toc117979550)

[Задание 6 16](#_Toc117979551)

[1.1. Постановка задачи 16](#_Toc117979552)

[1.2. Решение задачи, код программы 16](#_Toc117979553)

[1.3. Тестирование работы программы с проверкой 16](#_Toc117979554)

[Задание 7 21](#_Toc117979555)

[1.1. Постановка задачи 21](#_Toc117979556)

[1.2. Решение задачи, код программы 21](#_Toc117979557)

[1.3. Тестирование работы программы с проверкой 21](#_Toc117979558)

[2.2. Решение задачи, код программы 22](#_Toc117979559)

[2.3. Тестирование работы программы с проверкой 23](#_Toc117979560)

[Задание 8 25](#_Toc117979561)

[1.1. Постановка задачи 25](#_Toc117979562)

[1.2. Решение задачи, код программы 25](#_Toc117979563)

[1.3. Тестирование работы программы с проверкой 25](#_Toc117979564)

[Задание 9 26](#_Toc117979565)

[1.1. Постановка задачи 27](#_Toc117979566)

[1.2. Решение задачи, код программы 27](#_Toc117979567)

[1.3. Тестирование работы программы с проверкой 28](#_Toc117979568)

# Задание 1

## 1.1. Постановка задачи

Задание № 1. Ввести массив, состоящий из N элементов целого типа. Массив является элементами целого числа в двоичной системе счисления. Проверить правильность введения элементов двоичного числа, т.е. в массиве должны быть только 0 и 1. Если число введено верно, то перевести его в десятичную систему счисления и поэлементно записать число в массив. Пример:

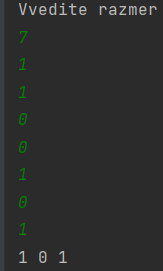
|  |  |
| --- | --- |
| В 2-ой | В 10-ой |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | 1 | 1 | 1 | 1 | 1 | | |  |  | | --- | --- | | 3 | 1 | |

## 1.2. Решение задачи, код программы

public class main8\_1 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Vvedite razmer");  
 int size = in.nextInt();  
 int[] array = new int[size];  
 for (int i = 0; i < size; i++) {  
 array[i] = in.nextInt();  
 if (array[i] < 0 || array[i] > 1) {  
 System.*out*.println("Nepravilno");  
 System.*exit*(0);  
 }  
 }  
 int sum = 0;  
 for (int i = 0; i < size; i++) {  
 int K = size - i - 1;  
 int num\_k = array[K];  
 sum += num\_k \* Math.*pow*(2, i);  
 }  
 int c =0;  
 int t = sum;  
 while ( t >0){  
 int k = t%10;  
 t = t/10;  
 c = c+1;  
 }  
 int[] array2 = new int[c];  
 int y = 0;  
 while ( sum%10 >0){  
 int r = y;  
 int l = (int) (sum / Math.*pow*(10,c-1));  
 array2[r] = l;  
 sum = (int) (sum % Math.*pow*(10,c-1));  
 c = c-1;  
 y+=1;  
 System.*out*.print(array2[r]);  
 System.*out*.print(" ");  
 }  
  
 }  
 }

}

## 1.3. Тестирование работы программы



# 

# Задание 2

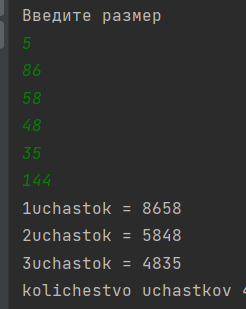
## 1.1. Постановка задачи

 Задан целочисленный массив. Определить количество участков массива, на котором элементы монотонно убывают (каждое следующее число меньше предыдущего), ), вывести их на экран в виде № участка – участок.

## 1.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class main8\_2 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите размер");  
 int size = in.nextInt();  
 int array[] = new int[size];  
 for (int i = 0; i <= array.length - 1; i++) {  
 array[i] = in.nextInt();  
 }  
 int k = 0;  
 for (int i = 0; i < size - 1; i++) {  
 if (array[i+1] < array[i]){  
 System.*out*.println(i+1+"uchastok = "+ array[i]+""+ array[i+1]);  
 }  
 k++;  
 }  
 int r = k;  
 System.*out*.println("kolichestvo uchastkov " + r);  
 }  
  
}

## 1.3. Тестирование работы программы



# Задание 3

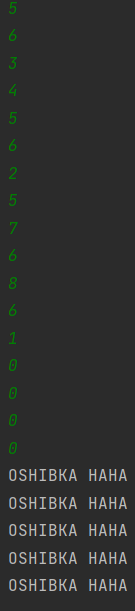
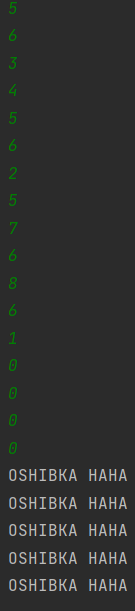
## 1.1. Постановка задачи

 В одномерном массиве с четным количеством элементов (2N) находятся координаты N точек плоскости. Они располагаются в следующем порядке: x1, y1, х2, y2, x3, y3, и т.д. Пользователем введены координаты центра и радиус окружности 1 (внутренний радиус кольцевой области) , , и координаты центра и радиус окружности 2 (внешний радиус кольцевой области) , , Кольцевая область определена на координатной плоскости. Определить номера точек, которые лежат вне кольца

## 1.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class main8\_3 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 int n = scanner.nextInt();  
 double[] coordn = new double[2\*n];  
 for (int i = 0; i < 2\*n; i++) {  
 coordn[i] = scanner.nextDouble();  
 }  
 double xc1 = scanner.nextDouble();  
 double yc1 = scanner.nextDouble();  
 double r1 = scanner.nextDouble();  
 double xc2 = scanner.nextDouble();  
 double yc2 = scanner.nextDouble();  
 double r2 = scanner.nextDouble();  
 double innerRadius = r2 - r1;  
 double outerRadius = r2;  
 for (int i = 0; i < 2\*n; i += 2) {  
 double x = coordn[i];  
 double y = coordn[i+1];  
 double d1 = Math.*sqrt*(Math.*pow*(x-xc1, 2) + Math.*pow*(y-yc1, 2));  
 double d2 = Math.*sqrt*(Math.*pow*(x-xc2, 2) + Math.*pow*(y-yc2, 2));  
 if (d1 < innerRadius && d2 > outerRadius) {  
 System.*out*.println("Точка " + (i/2 + 1) + " лежит вне кольца");  
 } else{  
 System.*out*.println("OSHIBKA HAHA");  
 }  
 }  
 }  
}

## 1.3. Тестирование работы программы



# Задание 4

## 1.1. Постановка задачи

 Задать массив из произвольных чисел и реализовать алгоритм сортировки массива по возрастанию методом прямого выбора: очевидно, что первое место в массиве должен занять минимальный элемент массива, второе – наименьший из всех остальных, третий – наименьший из оставшихся и т.д.

Для этого необходимо выполнить следующую последовательность действий:

* Просматривая массив от первого элемента, найти минимальный элемент и поместить его на место первого элемента, а первый – на место минимального.
* Просматривая массив от второго элемента, найти минимальный элемент и поместить его на место второго элемента, а второй – на место минимального.
* И так далее до предпоследнего элемента.

Выполнить сравнение сортировки методом прямого выбора с сортировкой «пузырек», рассмотрев количество итераций, а так же промежуточные результаты сортировки.

## 1.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class main8\_4 {  
public static void selectionSort(int[] sortArr) {  
 for (int i = 0; i < sortArr.length; i++) {  
 int pos = i;  
 int min = sortArr[i];  
 for (int j = i + 1; j < sortArr.length; j++) {  
 if (sortArr[j] < min) {  
 pos = j;  
 min = sortArr[j];  
 }  
 }  
 sortArr[pos] = sortArr[i];  
 sortArr[i] = min;  
 }  
 }  
  
public static void main(String args[]) {  
 Scanner in = new Scanner(System.*in*);  
 int N = in.nextInt();  
 int[] sortArr = new int[N];  
 for (int i = 0; i <= sortArr.length - 1; i++){  
 sortArr[i] = in.nextInt();  
}  
 *selectionSort*(sortArr);  
 for(int i = 0; i < sortArr.length; i++){  
 System.*out*.print(sortArr[i] + "\n");  
 }  
}  
}

## 1.3. Тестирование работы программы

## 

# Задание 5

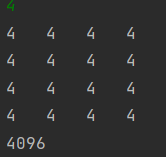
## 1.1. Постановка задачи

 Дан целочисленный массив В [n; n]. Вычислить произведение элементов этого массива, расположенных ниже побочной диагонали.

## 1.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class main8\_5 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("vvedite razmer");  
 int N = in.nextInt();  
 int[][] array = new int[N][N];  
 for (int i = 0; i < N; i++) {  
 for (int g = 0; g < N; g++) {  
 array[i][g] = N;  
 }  
 }  
 int k=0;  
 int s =1;  
 int p = 0;  
 for (int i = 1; i < N; i++) {  
 for ( int j =N - i;j < N; j++){  
 k = array[i][j];  
 p = (int) Math.*pow*(k,s);  
 s++;  
 }  
 }  
 for (int i = 0; i < N; i++) {  
 for (int g = 0; g < N; g++) {  
 System.*out*.print( array[i][g]+"\t");  
  
 }  
 System.*out*.println();  
 }  
 System.*out*.println(p);  
  
 }  
}

## 1.3. Тестирование работы программы



# Задание 6

## 1.1. Постановка задачи

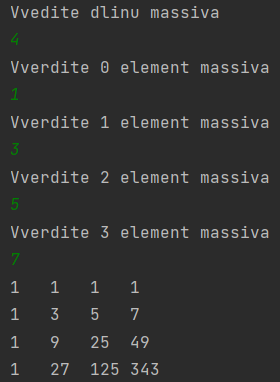
 Дан линейный массив x1, х2, ..., хn-1, хn. Получить действительную квадратную матрицу порядка n:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | … | 1 | 1 |
| x1 | x2 | … | xn-1 | xn |
| x12 | x22 | … | xn-12 | xn2 |
| … | … | … | … | … |
| x1n-1 | x2n-1 | … | xn-1n-1 | xnn-1 |

## 1.2. Решение задачи, код программы

import java.util.Scanner;  
  
public class main8\_6{  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Vvedite dlinu massiva" );  
 int size = in.nextInt();  
 int array [] = new int[size];  
 for ( int i = 0 ; i < size; i++){  
 System.*out*.println("Vverdite " + i + " element massiva");  
 array[i] = in.nextInt();  
 }  
 int [][]array2 = new int[size][size];  
 for ( int i = 0 ; i < size; i++){  
 for ( int g = 0; g< size; g++){  
 array2[i][g] = (int)Math.*pow*(array[g],i);  
 }  
 }  
 for ( int i = 0 ; i< array2.length; i++){  
 for ( int g = 0; g < array2.length; g++){  
 System.*out*.print( array2[i][g] +"\t");  
 }  
 System.*out*.println();  
 }  
  
  
 }  
  
}

## 1.3. Тестирование работы программы



# Задание 7

## Постановка задачи

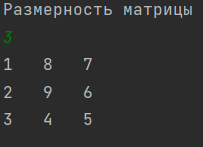
Заполнить двумерный массив n×n по образцу. Образец заполнения и пример:



## 1.2. Решение задачи, код программы

import java.util.Scanner;  
public class main8\_7 {  
 public static void main(String args[]) {  
 Scanner scn = new Scanner(System.*in*);  
 System.*out*.println("Размерность матрицы");  
 int N = scn.nextInt();  
 int[][] mas2 = new int[N][N];  
 *Zmey*(mas2,N);  
 *printMas2*(mas2);  
 }  
  
 public static int[][] Zmey(int[][] mas2, int N) {  
 int k = 0;  
 int sch = 1;  
 while ( k <= Math.*ceil*(N/2)){  
 for (int i = k; i < N - 1-k; i++) {  
 mas2[i][k] = sch++;  
 }  
 for (int i = k; i <= N - 1-k; i++) {  
 mas2[N-1-k][i] = sch++;  
 }  
 for (int i = N-2-k; i >= k; i--) {  
 mas2[i][N-1-k] = sch++;  
 }  
 for (int i = N-2-k ; i > k; i--) {  
 mas2[k][i] = sch++;  
 }  
 k=k+1;  
 }  
 return mas2;  
 }  
 public static void printMas2(int[][] array) {  
 for (int i=0; i<=array.length-1; i++) {  
 for (int j=0; j<=array[0].length-1; j++) {  
 System.*out*.print(array[i][j]+"\t");  
 }  
 System.*out*.println();  
 }  
 }  
}

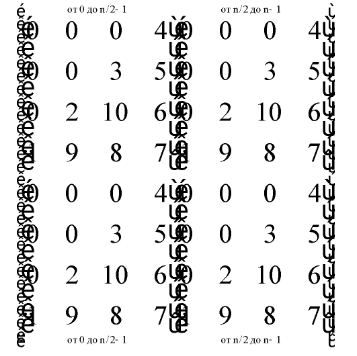
## 1.3. Тестирование работы программы



# Задание 8

## 1.1. Постановка задачи

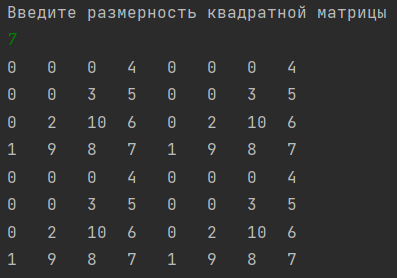
 Заполнить двумерный массив n×n по образцу. Матрица заполняется автоматически для любого n. Создать метод заполнения блока матрицы и отпараметризировав заполнение заполнить при помощи метода всю матрицу. Образец заполнения:



## 1.2. Решение задачи, код программы

import java.util.Scanner;  
public class main888\_8 {  
 public static void main(String args[]) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите размерность квадратной матрицы");  
 int N = in.nextInt();  
 if ( N%2 == 1){  
 N = N+1;  
 }  
 int A[][] = new int[N][N];  
  
 *Zmeika*(A,N,0,0);  
 *Zmeika*(A,N,0,N/2);  
 *Zmeika*(A,N,N/2,0);  
 *Zmeika*(A,N,N/2,N/2);  
 *printMas*(A);  
 }  
 public static void printMas(int array[][]){  
 for(int i = 0; i < array.length; i++){  
 for (int j = 0; j < array[0].length; j++){  
 System.*out*.print(array[i][j]+"\t");  
 }  
 System.*out*.println();  
 }  
 }  
 public static int [][] Zmeika(int[][]A,int N, int movei, int movej){  
 int k = 0, sch = 1;  
 while(k<= Math.*ceil*(N/2)){  
 for(int i = N/2-1-k; i >= 2\*k; i--){  
 A[i+movei][(N/2-1)+k-i+movej] = sch++;  
 }  
 for(int i = 1+2\*k; i <= N/2-1-k;i++){  
 A[i+ movei][N/2-1-k + movej] = sch++;  
 }  
 for(int i = N/2-2-k; i >= 1+2\*k;i--){  
 A[N/2-1-k + movei][i+ movej] = sch++;  
 }  
  
 k++;  
 }  
 return A;  
 }  
}

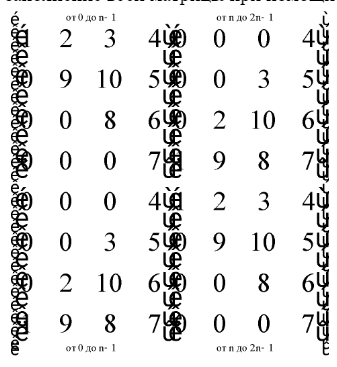
## 1.3. Тестирование работы программы



Задание 9

## 1.1. Постановка задачи

Заполнить двумерный массив 2n×2n по образцу. Матрица заполняется автоматически для любого n. Создать метод заполнения блоков матрицы и отпараметризировав выполнить заполнение всей матрицы при помощи методов. Образец заполнения:



## 1.2. Решение задачи, код программы

import java.util.Scanner;  
public class main8\_9 {  
 public static void main(String args[]) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Введите размерность квадратной матрицы");  
 int N = in.nextInt();  
 int A[][] = new int[2\*N][2\*N];  
 *Zmeika2*(A,N,0,0);  
 *Zmeika*(A,N,0,N);  
 *Zmeika*(A,N,N,0);  
 *Zmeika2*(A,N,N,N);  
 *printMas*(A);  
 }  
 public static void printMas(int array[][]){  
 for(int i = 0; i < array.length; i++){  
 for (int j = 0; j < array[0].length; j++){  
 System.*out*.print(array[i][j]+"\t");  
 }  
 System.*out*.println();  
 }  
 }  
 public static int [][] Zmeika(int[][]A,int N, int movei, int movej){  
 int k = 0, sch = 1;  
 while(k<= Math.*ceil*(N/2)){  
 for(int i = N-1-k; i >= 2\*k; i--){  
 A[i+movei][(N-1)+k-i+movej] = sch++;  
 }  
 for(int i = 1+2\*k; i <= N-1-k;i++){  
 A[i+ movei][N-1-k + movej] = sch++;  
 }  
 for(int i = N-2-k; i >= 1+2\*k;i--){  
 A[N-1-k + movei][i+ movej] = sch++;  
 }  
  
 k++;  
 }  
 return A;  
 }  
 public static int [][] Zmeika2(int[][]A,int N, int movei, int movej){  
 int k = 0, sch = 1;  
 while(k<= Math.*ceil*(N/2)){  
 for(int i = 2\*k; i <=N-1-k; i++) {  
 A[ k+ movei ][i + movej ] = sch++;  
 }  
 for(int i = k+1; i <=N-1-2\*k; i++) {  
 A[ i + movei][N-1-k + movej] = sch++;  
 }  
 for( int i = N-2-k ; i > 2\*k; i--){  
 A[i -k+ movei][i+ movej]= sch++;  
  
 }  
  
 k++;  
 }  
 return A;  
 }  
}

## 1.3. Тестирование работы программы

