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 20

Выполнил: студент группы ИСТ-22-1б Радостев К.П.

Пермь, 2023

**Содержание**

[Задание 1 4](#_Toc130423543)

[1.1. Постановка задачи 4](#_Toc130423544)

[1.2. Решение задачи, код программы 4](#_Toc130423545)

[1.3. Тестирование работы программы 4](#_Toc130423546)

[Задание 2 6](#_Toc130423547)

[2.1. Постановка задачи 6](#_Toc130423548)

[2.2. Решение задачи, код программы 6](#_Toc130423549)

[2.3. Тестирование работы программы 7](#_Toc130423550)

[Задание 3 8](#_Toc130423551)

[3.1. Постановка задачи 8](#_Toc130423552)

[3.2. Решение задачи, код программы 8](#_Toc130423553)

[3.3. Тестирование работы программы с проверкой 9](#_Toc130423554)

[Задание 4 10](#_Toc130423555)

[4.1. Постановка задачи 10](#_Toc130423556)

[4.2. Решение задачи, код программы 10](#_Toc130423557)

[4.3. Тестирование работы программы 11](#_Toc130423558)

[Задание 5 12](#_Toc130423559)

[5.1. Постановка задачи 12](#_Toc130423560)

[5.2. Решение задачи, код программы 12](#_Toc130423561)

[5.3. Тестирование работы программы 13](#_Toc130423562)

[Задание 6 14](#_Toc130423563)

[6.1. Постановка задачи 14](#_Toc130423564)

[6.2. Решение задачи, код программы 14](#_Toc130423565)

[6.3. Тестирование работы программы с проверкой 15](#_Toc130423566)

[Задание 7 16](#_Toc130423567)

[7.1. Постановка задачи 16](#_Toc130423568)

[7.2. Решение задачи, код программы 16](#_Toc130423569)

[7.3. Тестирование работы программы 16](#_Toc130423570)

[Задание 8 18](#_Toc130423571)

[8.1. Постановка задачи 18](#_Toc130423572)

[8.2. Решение задачи, код программы 18](#_Toc130423573)

[8.3. Тестирование работы программы 19](#_Toc130423574)

[Задание 9 20](#_Toc130423575)

[9.1. Постановка задачи 20](#_Toc130423576)

[9.2. Решение задачи, код программы 20](#_Toc130423577)

[9.3. Тестирование работы программы с проверкой 21](#_Toc130423578)

[Задание 10 22](#_Toc130423579)

[10.1. Постановка задачи 22](#_Toc130423580)

[10.2. Решение задачи, код программы 22](#_Toc130423581)

[10.3. Тестирование работы программы с проверкой 23](#_Toc130423582)

[Задание 11 24](#_Toc130423583)

[11.1. Постановка задачи 24](#_Toc130423584)

[11.2. Решение задачи, код программы 24](#_Toc130423585)

[11.3. Тестирование работы программы с проверкой 26](#_Toc130423586)

[Задание 12 27](#_Toc130423587)

[12.1. Постановка задачи 27](#_Toc130423588)

[12.2. Решение задачи, код программы 27](#_Toc130423589)

[12.3. Тестирование работы программы с проверкой 28](#_Toc130423590)

# Задание 1

## Постановка задачи

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 шестеричную систему счисления и поэлементно записать число в массив.

## Решение задачи, код программы

1. import java.util.Scanner;  
   public class Zad1 {  
    public static void main(String[] args) {  
    Scanner mc = new Scanner(System.*in*);  
    System.*out*.println("Введите длину числа 10 is MAX");  
    int m = mc.nextInt();  
    int mas[] = new int[m];  
    String chis = "";  
    for (int i = 0; i <= m - 1; i++) {  
    System.*out*.print("n" + i + " = ");  
    int n = 0;  
    int j = 0;  
    while (j != 1) {  
    System.*out*.print("");int b = mc.nextInt();  
    if (b == 1 || b == 0) {  
    j += 1;  
    n = b;  
    chis += n;  
    }  
    }  
    mas[i] = n;  
    }  
    int i = Integer.*parseInt*(chis);  
    int desCh = Integer.*parseInt*(chis, 2);  
    int mas2[] = new int[m];  
    int h = 0;  
    while (desCh > 0) {  
    mas2[h] = desCh % 6;  
    desCh /= 6;  
    h++;  
    }  
    *prmas*(mas2);  
    }  
    public static void prmas(int [] mas){  
    for (int i=mas.length-1;i>=0;i--){  
    if (mas[i]!=0){  
    System.*out*.print(mas[i]);  
    }  
     
    }  
    }  
   }

## 1.3. Тестирование работы программы

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Решение онлайн-калькулятора |
| 1 |  |  |
| 2 |  |  |

Сравнение решения задачи с использованием двух прикладных пакетов показала, что решения задачи в Java и онлайн-калькулятора совпадает. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

Задание 2

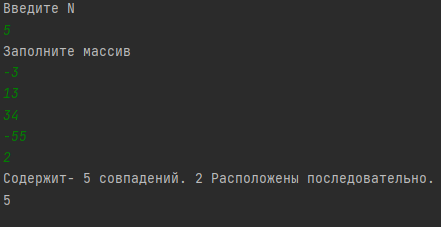
## 2.1. Постановка задачи

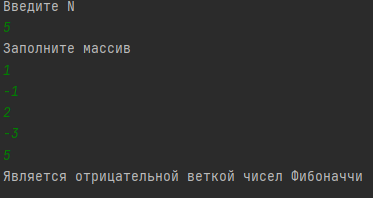
. Последовательность чисел Фибоначчи FK описывается следующими формулами:  Иногда числа Фибоначчи рассматривают и для отрицательных значений n как двусторонне бесконечную последовательность. При этом члены с отрицательными индексами легко получить с помощью эквивалентной формулы: Пользователь вводит одномерный массив размерностью N. Определить является ли массив отрицательной веткой последовательности Фибоначчи в явном виде, т.е. элементы стоят в правильном порядке по убыванию, или массив содержит все необходимые числа отрицательной ветки чисел Фибоначчи, но они стоят не в том порядке, либо процент чисел соответствующих последовательности Фибоначчи (отрицательной ветке).

## 2.2. Решение задачи, код программы

import java.util.Scanner;  
  
import static java.lang.Math.*pow*;  
  
public class work\_2 {  
 public static void main(String[] args) {  
 Scanner mc = new Scanner(System.*in*);  
 System.*out*.println("Введите N");  
 int N = mc.nextInt();  
 int[] mas\_User = new int[N+1];  
 int[] mas\_POL = new int[10 \* N];  
 int[] mas\_Otr = new int[10 \* N];  
 System.*out*.println("Заполните массив");  
 for (int i = 2; i < 10 \* N; i++) {  
 mas\_POL[0] = 1;  
 mas\_POL[1] = 1;  
 mas\_POL[i] = mas\_POL[i - 2] + mas\_POL[i - 1];  
 }  
 for (int i = 0; i < 10 \* N; i++) {  
 mas\_Otr[i] = (int) (*pow*(-1, i + 2) \* mas\_POL[i]);  
 }  
 int i = 0;  
 for (i = 0; i < N; i++) {  
 mas\_User[i] = mc.nextInt();  
 }  
 int j = 0, k = 1, m = 0, chis = 0;  
 for (i = 0; i <N;) {  
 for (j = 0; j < 10 \* N; j++) {  
 if (mas\_User[i] == mas\_Otr[j]) {  
 chis++;  
 if (mas\_Otr[j]==mas\_User[i] & mas\_User[i+1]==mas\_Otr[j+1]) {  
 k++;  
 if (k > m) {  
 m = k;  
 }  
 }  
 }  
 }i++;j=0;  
 }  
 if (chis == k) {  
 System.*out*.println("Является отрицательной веткой чисел Фибоначчи");}  
 else System.*out*.println("Содержит- " + chis + " совпадений. "+ k +" Расположены последовательно.");  
 }  
}

## 2.3. Тестирование работы программы





# Задание 3

## 3.1. Постановка задачи

В одномерном массиве с четным количеством элементов (2N) находятся координаты N точек плоскости. Они располагаются в следующем порядке: и т.д. Найти номера самых удаленных друг от друга точек и наименее удаленных друг от друга точек.

## 3.2. Решение задачи, код программы

import java.util.Random;  
import java.util.Scanner;  
  
import static java.lang.System.*out*;  
  
public class Zad3 {  
 public static void main(String[] args) {  
 Scanner mc = new Scanner(System.*in*);  
 System.*out*.println("vvedi n");  
 int n = mc.nextInt();  
 int result = 1;  
 for (int i = 1; i <= n; i++) {  
 result \*= i;  
 }  
 Random ran = new Random();  
 int mas\_1[] = new int[2 \* n];  
 for (int i = 0; i < 2 \* n; i++) {  
 mas\_1[i] = ran.nextInt(7,66);  
 }  
 int k = (int) Math.*sqrt*(Math.*pow*((mas\_1[3] - mas\_1[1]),2) + Math.*pow*((mas\_1[2] - mas\_1[0]),2));  
 int k1=k;  
 int x11=0, y11=0, x22=0, y22=0, x1=0, y1=0, x2=0, y2=0;  
 for (int i = 0; i < 2 \* n; i += 2) {  
 for (int j = i+2; j < 2 \* n; j += 2) {  
 if (Math.*sqrt*(Math.*pow*((mas\_1[j] - mas\_1[i]), 2) + Math.*pow*((mas\_1[j + 1] - mas\_1[i + 1]), 2)) > k) {  
 k = (int) (Math.*sqrt*(Math.*pow*((mas\_1[j] - mas\_1[i]), 2) + Math.*pow*((mas\_1[j + 1] - mas\_1[i + 1]), 2)));  
 x1 = mas\_1[i];  
 y1 = mas\_1[i + 1];  
 x2 = mas\_1[j];  
 y2 = mas\_1[j + 1];  
 }  
 }  
 }  
 for (int i = 0; i < 2 \* n; i += 2) {  
 for (int j = i; j < 2 \* n; j += 2) {  
 if (Math.*sqrt*(Math.*pow*((mas\_1[j] - mas\_1[i]), 2) + Math.*pow*((mas\_1[j + 1] - mas\_1[i + 1]), 2)) < k1 & mas\_1[i]!=mas\_1[j] & mas\_1[j+1]!=mas\_1[i+1]) {  
 k1 = (int) (Math.*sqrt*(Math.*pow*((mas\_1[j] - mas\_1[i]), 2) + Math.*pow*((mas\_1[j + 1] - mas\_1[i + 1]), 2)));  
 x11 = mas\_1[i];  
 y11 = mas\_1[i + 1];  
 x22 = mas\_1[j];  
 y22 = mas\_1[j + 1];  
 }  
 }  
 }  
 System.*out*.println(x1 + " " + y1 + " " + x2 + " " + y2 + " расстояние = "+ k);  
 System.*out*.println(x11 + " " + y11 + " " + x22 + " " + y22 +" расстояние = "+ k1);  
 *printmas\_1*(mas\_1);  
  
 }  
 public static void printmas\_1(int[] array) {  
 for (int i=0; i<=array.length-1; i++) {  
 *out*.print(array[i]+"\t");  
 }  
 }  
}

## 3.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в ячейку в диапазоне А2:F2 записаны точки, расстояние между которых нужно вычислить, в ячейке А4 и B4 формулы максимума и минимума.Формулы:

H2=КОРЕНЬ((E2-C2)^2+(F2-D2)^2)

I2=КОРЕНЬ((E2-A2)^2+(F2-B2)^2)

J2=КОРЕНЬ((C2-A2)^2+(D2-B2)^2)

А4 =МАКС(H2:J2)

В4 =МИН(H2:J2)

Далее в таблице 1 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Решение MS Excel |
| 1 |  |  |
| 2 |  |  |

Сравнение решения задачи с использованием двух прикладных пакетов показала, что решения задачи в Java и MS Excel совпадает. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.

Задание 4

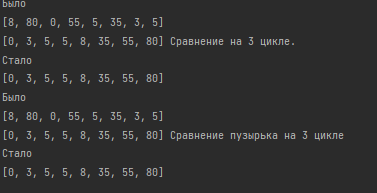
## 4.1. Постановка задачи

Алгоритм быстрой сортировки. Отличительной особенностью быстрой сортировки является операция разбиения массива на две части относительно опорного элемента. Пусть крайний левый элемент опорный pivot. Установим указатель left на следующий за ним элемент; right – на последний. Алгоритм должен определить правильное положение элемента pivot. Указатель left движется вправо, пока элементы, на которые он показывает, остаются меньше опорного. Указатель right движется влево, пока элементы, на которые он показывает, остаются больше опорного. Указатель left перемещается до тех пор, пока не покажет элемент больше pivot; right движется, пока не покажет элемент меньше pivot. Элементы left и right меняются местами. И движением возобновляется снова. Процесс продолжается до тех пор, пока right не окажется слева от left. Тем самым будет определено правильное место опорного элемента. Осуществляется перестановка опорного элемента с элементом, на который указывает right. Опорный элемент находится в нужном месте: элементы слева от него имеют меньшие значения; справа – большие. Алгоритм рекурсивно вызывается для сортировки подмассивов слева от разрешающего и справа от него. Реализовать алгоритм быстрой сортировки. Выполнить сравнение быстрой сортировки с сортировкой «пузырек», рассмотрев количество итераций, а так же промежуточные результаты сортировки.

## 4.2. Решение задачи, код программы

import java.util.Arrays;  
  
public class Zad4 {  
 public static void main(int[] mas,int niz, int verh) {  
 int k=0;  
 if (mas.length==0)  
 return;  
 if(niz>=verh)  
 return;  
 int seredina=niz+(verh-niz)/2;  
 int pivot=mas[seredina];  
 int i=niz,j=verh;  
 while (i<=j) {  
 k++;  
 while (mas[i] < pivot) {  
 i++;  
 }  
 while (mas[j] > pivot) {  
 j--;  
 }  
 if (i <= j) {  
 int temp = mas[i];  
 mas[i] = mas[j];  
 mas[j] = temp;  
 i++;  
 j--;  
 }  
 }  
 if (niz<j)  
 *main*(mas,niz,j);  
 if (verh>i)  
 *main*(mas,i,verh);  
 if (k==3) System.*out*.println(Arrays.*toString*(mas)+" Сравнение на 3 цикле.");  
 }  
 public static void main(String[] args) {  
 int[] x = {8,80,0,55,5,35,3,5};  
 int[] y = {8,80,0,55,5,35,3,5};  
 int counter = 0; int temp; int low = 0; int high = x.length - 1;  
 System.*out*.println("Было");  
 System.*out*.println(Arrays.*toString*(x));  
 *main*(x, low, high);  
 System.*out*.println("Стало");  
 System.*out*.println(Arrays.*toString*(x));  
 boolean isSorted = false; // Пузырек  
 System.*out*.println("Было");  
 System.*out*.println(Arrays.*toString*(y));  
 while (!isSorted) {  
 isSorted = true;  
 for (int j = 0; j < x.length - 1; j++) {  
 if (y[j] > y[j + 1]) {  
 temp = y[j];  
 y[j] = y[j + 1];y[j + 1] = temp;  
 isSorted = false;  
 counter++;  
 if (counter == 3) System.*out*.println(Arrays.*toString*(x) + " Cравнение пузырька на 3 цикле");  
 }  
 }  
 }  
 System.*out*.println("Стало");  
 System.*out*.println(Arrays.*toString*(y));  
 }  
}

## 4.3. Тестирование работы программы



# Задание 5

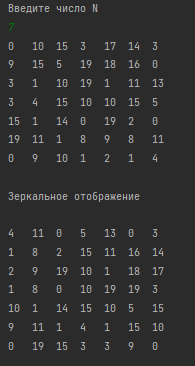
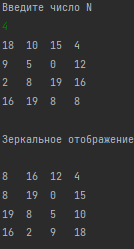
## 5.1. Постановка задачи

Дана квадратная матрица А порядка М. Зеркально отразить ее элементы относительно побочной диагонали (при этом элементы побочной диагонали останутся на прежнем месте, элемент  поменяется местами с , элемент  - c и т.д.). Вспомогательную матрицу не использовать.

## 5.2. Решение задачи, код программы

import java.util.Random;  
import java.util.Scanner;  
public class Zad5 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Введите число N");  
 int n = scanner.nextInt();  
 Random ran = new Random();  
 int[][] mas\_1 = new int[n][n];  
 for (int i = 0; i < mas\_1.length; i++) {  
 for (int g = 0; g < mas\_1[i].length; g++) {  
 mas\_1[i][g] = ran.nextInt(20);  
 }  
 }  
 for (int i = 0; i < mas\_1[0].length; i++) {  
 for (int[] ints : mas\_1) {  
 System.*out*.print(ints[i] + "\t");  
 }  
 System.*out*.println();  
 }  
 System.*out*.println("");  
 for (int i=0;i< mas\_1[0].length;i++){ //Зеркало  
 for (int g=0;g< mas\_1.length;g++){  
 System.*out*.print(mas\_1[n-1-i][n-1-g]+"\t");  
 }  
 System.*out*.println("");  
 }  
 }  
}

## 5.3. Тестирование работы программы

Задание 6

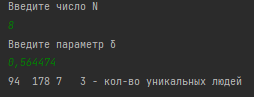
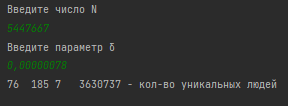
## 6.1. Постановка задачи

Пусть целочисленная матрица размером n × m содержит информацию об учениках некоторого класса из n человек. В 0-м столбце проставлена масса (кг), во 1-ом – рост (см), в 2-ом – успеваемость (средний балл), в 3-ем – численность семьи. Ученик называется среднестатистическим по k-му параметру (уникальным по k-му параметру), если на нем достигается минимум (максимум) модуля разности среднего арифметического чисел из k-го столбца и значения k-го параметра этого ученика. Определить количество среднестатистических учеников в классе по численности семьи (минимум модуля разности среднего арифметического чисел из k-го столбца и значения k-го параметра этого ученика  (параметр веденный пользователем, бесконечно малое)) и записать их в отдельный двумерный массив и самого уникального ученика по численности семьи и записать его данные в одномерный массив. Создать уникальный стилизованный вывод результатов.

## 6.2. Решение задачи, код программы

import java.util.Random;  
import java.util.Scanner;  
  
public class Zad6 {  
 public static void main(String[] args) {  
 Scanner ms = new Scanner(System.*in*);  
 System.*out*.println("Введите число N");  
 int n = ms.nextInt();  
 System.*out*.println("Введите параметр δ ");  
 double m = ms.nextDouble();  
 int[][] mas\_2 = new int[n][3];  
 Random ran = new Random();  
 for (int i = 0; i < n; i++) {  
 mas\_2[i][0] = ran.nextInt(40, 100);  
 mas\_2[i][1] = ran.nextInt(120, 200);  
 mas\_2[i][2] = ran.nextInt(2, 8);  
 }  
 double k = 0;int[] Unik=new int[3];int Uni=0;  
 for (int i = 0; i < n; i++) {  
 if (mas\_2[i][2]>Uni){  
 Uni=mas\_2[i][2];  
 Unik[0] = mas\_2[i][0];  
 Unik[1] = mas\_2[i][1];  
 Unik[2] = mas\_2[i][2];  
 }  
 }  
 int cnt=0,sum3=0;  
 for (int i =0;i<n;i++) {  
 sum3 += mas\_2[i][2];  
 }  
 double sr3=sum3/n;  
 for (int i = 0;i<n;i++){  
 if(mas\_2[i][2]+m>=sr3){  
 mas\_2[i][2]=1;  
 cnt++;  
 }  
 }  
 *printMas2*(Unik); System.*out*.println(cnt+" - кол-во уникальных людей");  
 }  
 public static void printMas2(int[] mas) {  
 for (int i=0; i<=mas.length-1; i++) {  
 System.*out*.print(mas[i]+"\t");  
 }  
 }  
}

## 6.3. Тестирование работы программы с проверкой

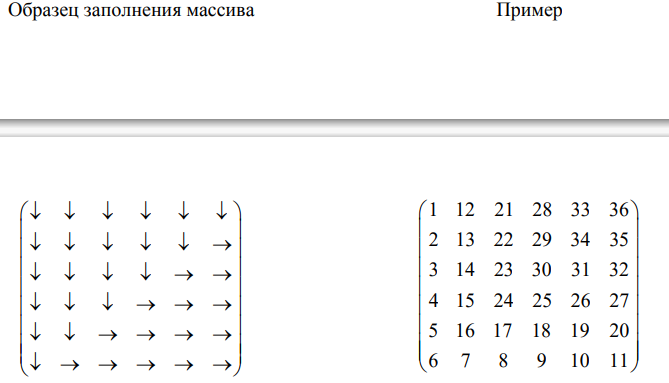
 

# Задание 7

## 7.1. Постановка задачи

Заполнить двумерный массив n×n по образцу.

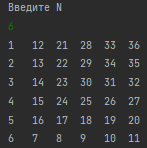
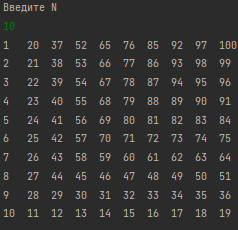
Образец заполнения и пример:



## 7.2. Решение задачи, код программы

import java.util.Scanner;  
public class Zad7 {  
 public static void main(String[] args) {  
 Scanner chis = new Scanner(System.*in*);  
 System.*out*.println("Введите N");  
 int n = chis.nextInt();  
 int[][] mas = new int[n][n];  
 int sch = 1;int m=n;  
 for (int i = 0; i < n; i++) {  
 for (int g = 0; g < m; g++) {  
 mas[g][i] = sch++;  
 if (g == m - 1) {  
 for (int a = i + 1; a < n; a++) {  
 mas[g][a] = sch++;  
 }  
 m--;  
 }  
 }  
 }  
 *printMas*(mas);  
 }  
 public static void printMas(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();  
 }  
 }  
}

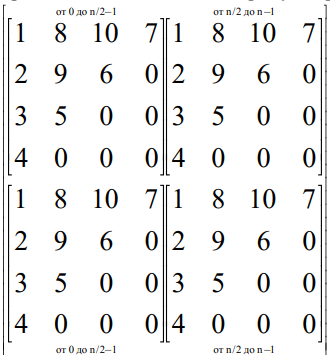
## 7.3. Тестирование работы программы

# Задание 8

## 8.1. 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:

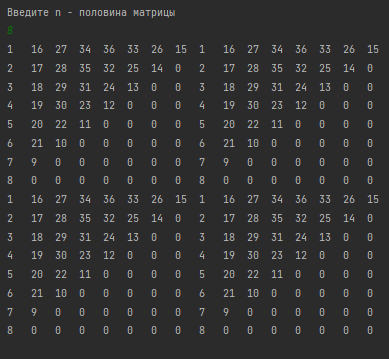


## 

## 8.2. Решение задачи, код программы

import static java.lang.System.*out*;  
import static java.lang.Math.\*;  
import java.util.Scanner;  
public class Zad8 {  
 public static void main(String args[]) {  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Введите n - половина матрицы");  
 int n = scan.nextInt();  
 int[][] myMas2 = new int[2 \* n][2 \* n];  
 *MMM*(myMas2,n,1);*MMM*(myMas2,n,2);*MMM*(myMas2,n,3);*MMM*(myMas2,n,4);  
 *printMas2*(myMas2);  
 }  
 public static void MMM(int[][] myMas2,int n,int pp){  
 int k = 0, sch = 1;  
 while (k <= *ceil*(n / 3)) {  
 for (int i = *numb*(pp,n); i <*numb1*(pp)\*n - 1; i++) {  
 int y = i + 1;  
 for (int x = *numb2*(pp,n); x < *numb3*(pp)\*n - i; x++) {  
 if (myMas2[x][i] == 0) {  
 myMas2[x][i] = sch++;  
 }  
 }  
 for (int x = i \* 2; x < *numb4*(pp)\*n - 1; x++) {  
 myMas2[pp\*n - x - 2][y] = sch++;  
 y++;  
 }k++;  
 }  
 }  
 }  
 public static int numb(int m,int n){  
 if (m==1||m==2){return 0;}  
 return n;}  
 public static int numb1(int n){  
 if(n==1||n==2){return 1;}  
 return 2;}  
 public static int numb2(int m,int n){  
 if (m ==2||m ==4){return n;}  
 return 0;}  
 public static int numb3(int n){  
 if (n ==2||n ==3){return 2;}  
 else if (n==4) {return 3;}  
 return 1;}  
 public static int numb4(int n){  
 if (n ==1||n ==2){return 1;}  
 return 3;}  
 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++) {  
 *out*.print(array[i][j]+"\t");  
 }  
 *out*.println();  
 }  
 }  
}

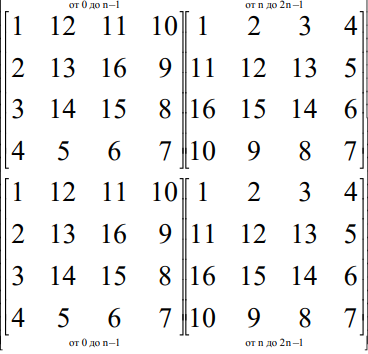
## 8.3. Тестирование работы программы

Задание 9

## 9.1. Постановка задачи

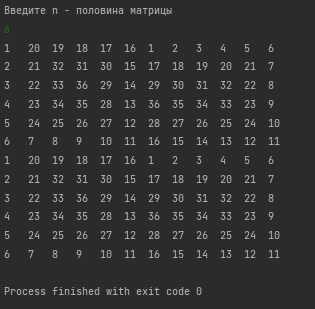
Заполнить двумерный массив 2n×2n по образцу. Матрица заполняется автоматически для любого n. Создать метод заполнения блоков матрицы и отпараметризировав выполнить заполнение всей матрицы при помощи методов. Образец заполнения: 



## 9.2. Решение задачи, код программы

import static java.lang.System.*out*;  
import static java.lang.Math.\*;  
import java.util.Scanner;  
public class Zad9 {  
 public static void main(String args[]){  
 Scanner scan = new Scanner (System.*in*);  
 System.*out*.println("Введите n - половина матрицы");  
 int n = scan.nextInt();  
 int [][] myMas2 = new int [2\*n][2\*n];  
 *MMM*(myMas2,n,1);*MMM*(myMas2,n,1);*MMM*(myMas2,n,2);*MMM*(myMas2,n,3);*MMM*(myMas2,n,4);  
 *printMas2*(myMas2);  
 }  
 public static void MMM(int [][]myMas2,int n,int pp){  
 if(pp==1||pp==2) {  
 int k = 0, minN = 0, sch = 1;  
 while (k <= *ceil*(n / 3)) {  
 for (int i = n; i < 2 \* n - minN; i++) {  
 myMas2[minN + *numb*(pp) \* n][i] = sch++;  
 }  
 for (int j = 1 + minN; j < n - minN; j++) {  
 myMas2[j + *numb*(pp)\*n][2 \* n - 1 - minN] = sch++;  
 }  
 for (int i = 2 \* n - 2 - minN; i >= n; i--) {  
 myMas2[*numb1*(pp) \* n - 1 - minN][i] = sch++;  
 }  
 k++;  
 minN++;  
 }  
 }  
 else{  
 int k=0,sch = 1,minN=0;  
 while (k<=*ceil*(n/3)) {  
 for (int i =*numb*(pp)\*n + minN; i<*numb1*(pp)\*n-minN; i++) {  
 myMas2[i][minN] = sch++;  
 }  
 for (int j = 1+minN; j<n-minN; j++) {  
 myMas2[*numb1*(pp)\*n-1- minN][j] = sch++;  
 }  
 for (int i=*numb1*(pp)\*n-2-minN; i>=*numb*(pp)\*n+minN; i--) {  
 myMas2[i][n-1-minN] = sch++;  
 }  
 for (int j=n-2-minN; j>minN; j--) {  
 myMas2[*numb*(pp)\*n+minN][j] = sch++;  
 }  
 k++;  
 minN++;  
 }  
 }  
 }  
 public static int numb(int n){  
 if (n==1||n==3){return 0;}  
 return 1;}  
 public static int numb1(int n){  
 if(n==1||n==3){return 1;}  
 return 2;}  
 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++) {  
 *out*.print(array[i][j]+"\t");  
 }  
 *out*.println();  
 }  
 }  
}

## 9.3. Тестирование работы программы с проверкой

Задание 10

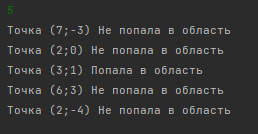
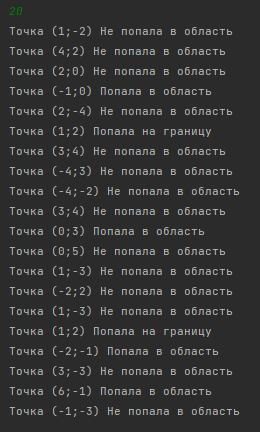
## 10.1. Постановка задачи

Дана матрица с N точками в пространстве, 0-я строчка координаты х, 1-я строчка координаты y. Определить для каждой точки попала ли она в закрашенную область, с использованием массива N элементов (1 – точка попала внутрь области, 0 – точка на Дана матрица с N точками в пространстве, 0-я строчка координаты х, 1-я строчка координаты y. Определить для каждой точки попала ли она в закрашенную область, с использованием массива N элементов (1 – точка попала внутрь области, 0 – точка на границе области, -1 – точка вне области). Области взять из лабораторной работы № 2 задание 5 (условия проверки реализовать как метод). Создать метод для стилизованного вывода ответов по попаданию точки в область в соответствии с массивом данных о попадании точки.

## 10.2. Решение задачи, код программы

import java.util.Scanner;  
import java.util.Random;  
public class Zad10 {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 Random run = new Random();  
 int n= sc.nextInt();  
 int[][] mas\_1 = new int[2][n];  
 for (int i=0;i<n;i++){//Заполнение массива автоматом//  
 mas\_1[0][i]= run.nextInt(-4,8);  
 mas\_1[1][i]=run.nextInt(-4,6);  
 }  
 for (int i=0;i<n;i++){  
 int pop=*FIGURE*(mas\_1[0][i],mas\_1[1][i]);  
 if (pop==1){  
 System.*out*.println("Точка ("+mas\_1[0][i]+";"+mas\_1[1][i]+") Попала в область");  
 }  
 if (pop==-1) {  
 System.*out*.println("Точка (" + mas\_1[0][i] + ";" + mas\_1[1][i] + ") Не попала в область");  
 }  
 if (pop==0) {  
 System.*out*.println("Точка (" + mas\_1[0][i] + ";" + mas\_1[1][i] + ") Попала на границу");  
 }  
 }  
 }  
 public static int FIGURE(int x, int y) {  
 if (*fig\_1*(x,y)==1){  
 return 1;  
 }  
 else if (*fig\_1*(x,y)==0) {  
 return 0;  
 }  
 else{  
 return -1;  
 }  
 }  
 public static int fig\_1(double x, double y) {  
 double v1 = Math.*pow*(x + 1, 2) + Math.*pow*(y - 1, 2), v2 = Math.*pow*(x - 4, 2) + Math.*pow*(y + 1, 2);  
 if (((x>=-1 & x<=1 & y>=-1 & y<=5 & y<-1.5\*x+3.5 & y>3\*x-1 & y>-2\*x-1)||  
 (y>=-1 & y<=3 & x>=-3 & x<=-1 & y<v1 & y>-v1))||(((x>=2 & x<=4 & y>=0 & y<=1 & y>-x+3)||(x>=2 & x<=3 & y<=0 & y>=-1 & y<x-3)||  
 (y>x-7 & x>=4 & x<=6 & y>=-3 & y<=-1)||(x>=2 & x<=4 & y>=-3 & y<=-1 & y<-v2)||  
 (x>=4 & x<=6 & y>=-1 & y<=1 & y<v2)))){return 1;}  
 if ((x>-1 & x<=1 & y>=2 & y<=5 & y==-1.5\*x+3.5)||(y==3\*x+1&y>=-1&y<=2&0<=x&x<=1)  
 ||(x>-1 & x<=0 & y<1 & y>=-1 & y==-2\*x-1)||(x==-1 & y>=-1 & y<=1)||(x==-1 & y>=3 & y<=5)||(x>=-3 & x<=-1 & y<=3 & y>=-1 & 4==v1)  
 ||(x>=2 & x<=4 & y<=-1 & y>=-3 & 4==v2)||(x>=4 & x<=6 & y<=1 & y>=-1 & 4==v2)||(x>=4 & x<=6 & y>=-3 & y<=-1 & y==x-7)||(y==1 & x>=2 & x<=4)  
 ||(x>=2 & x<=3 & y<=1& y>=0 & y==-x+3)||(x>=2 & x<=3 & y<=0 & y>=-1 & y==x-3)) {return 0;}  
 return -1;  
 }  
}

## 10.3. Тестирование работы программы с проверкой

Задание 11

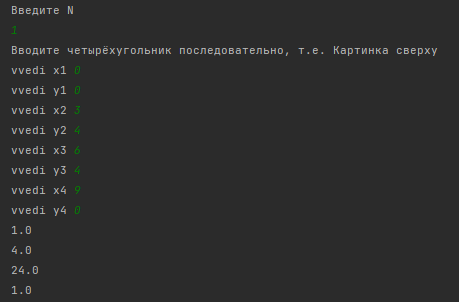
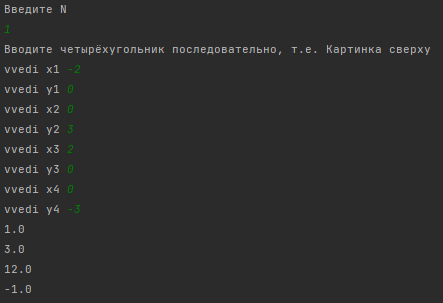
## 11.1. Постановка задачи

Дан двумерный массив 2×4N, где N – количество предполагаемых четырехугольников. Пользователь вводит данные по координатам вершин четырехугольников в двумерный массив. Создать двумерный массив ответов для N предполагаемых четырехугольников: 0-я строчка результаты проверки о существовании четырехугольника с введенными вершинами (1 – существует; 0 – не существует); 1-я строчка классификация четырехугольников (1 – прямоугольник; 2 – квадрат; 3 – ромб; 4 – трапеция; 5 – произвольный; 0 – если такого четырехугольника не существует); 2-я строчка площадь четырехугольника, если он существует, иначе 0; 3-я строчка сообщает результаты проверки можно ли вписать четырехугольник в окружность (-1 – если нельзя; 1 – если можно; 0 – если четырехугольника не существует). Создать метод для стилизованного вывода ответов.

## 11.2. Решение задачи, код программы

import java.util.Scanner;  
  
import static java.lang.Math.\*;  
import static java.lang.System.*out*;  
  
public class Zad11 {  
 public static void main(String[] args) {  
 Scanner mc = new Scanner(System.*in*);  
 *out*.println("Введите N");  
 int n = mc.nextInt(), cnt = 0; //2 //3  
 int[][] mas\_1 = new int[2][4 \* n];  
 double [][] mas\_con = new double[4][n]; //1 //4  
 System.*out*.println("Вводите четырёхугольник последовательно, т.е. Картинка сверху");  
 for (int i = 0; i <= 4 \* n - 1; i++) {  
 if (i % 4 == 0) {  
 cnt++;}  
 System.*out*.print("vvedi x" + (i % 4 + 1));  
 *out*.print(" ");mas\_1[0][i] = mc.nextInt();  
 System.*out*.print("vvedi y" + (i % 4 + 1));  
 *out*.print(" ");mas\_1[1][i] = mc.nextInt();  
 }  
// for(int i=0;i<n;i++){  
// for(int j=i;j<n;j++){  
// if(mas\_1[0][i]>mas\_1[0][j]){  
// int c = mas\_1[0][i], m= mas\_1[1][i];  
// mas\_1[0][i]=mas\_1[0][j]; что то не выходит сортировка :(((((  
// mas\_1[1][i]=mas\_1[1][j];  
// mas\_1[0][j]=c;mas\_1[1][j]=m;}  
// }  
// }  
 for (int i = 0; i < n; i++) {  
 double ab = *sqrt*((*pow*(mas\_1[0][i + 1] - mas\_1[0][i], 2) + *pow*(mas\_1[1][i + 1] - mas\_1[1][i], 2)));  
 double bc = *sqrt*((*pow*(mas\_1[0][i + 2] - mas\_1[0][i + 1], 2) + *pow*(mas\_1[1][i + 2] - mas\_1[1][i + 1], 2)));  
 double cd = *sqrt*((*pow*(mas\_1[0][i + 3] - mas\_1[0][i + 2], 2) + *pow*(mas\_1[1][i + 3] - mas\_1[1][i + 2], 2)));  
 double da = *sqrt*((*pow*(mas\_1[0][i] - mas\_1[0][i + 3], 2) + *pow*(mas\_1[1][i] - mas\_1[1][i + 3], 2)));  
  
 double ac = *sqrt*(*pow*(mas\_1[0][i + 2] - mas\_1[0][i], 2) + *pow*(mas\_1[1][i + 2] - mas\_1[1][i], 2));  
 double bd = *sqrt*(*pow*(mas\_1[0][i + 3] - mas\_1[0][i+1], 2) + *pow*(mas\_1[1][i + 3] - mas\_1[1][i+1], 2));  
  
 double cosBAD=(-*pow*(bd,2)+*pow*(ab,2)+*pow*(da,2))/(2\*ab\*da);//Для площади трапеции  
 double sinBAD=*sqrt*(1-(*pow*(cosBAD,2)));  
 double cosBCD=(-*pow*(bd,2)+*pow*(bc,2)+*pow*(cd,2))/(2\*bc\*cd);  
 double sinBCD=*sqrt*(1-(*pow*(cosBCD,2)));//  
  
 double m = Math.*toDegrees*(*acos*(cosBCD));double z=Math.*toDegrees*(*acos*(cosBAD));  
 if ((ab<bc+cd+da & bc<ab+cd+da & cd<bc+ab+da & da<bc+cd+ab)||(ab==bc & bc==cd & cd==da & da==ab)) {  
 mas\_con[0][i] = 1;}  
 else {mas\_con[0][i]=0;mas\_con[1][i]=0;mas\_con[2][i]=0;mas\_con[3][i]=0;}  
  
 if (ab == bc & bc == cd & cd == da & da == ab & ac == bd) {//"1) Существует ли? 2) какой четырехуголь-ник? 3) какова площадь? 4) можно ли вписать в окружность?"//  
 mas\_con[1][i] = 2; mas\_con[2][i]= (ab\*bc);mas\_con[3][i]=1;}  
 else if (ab == bc & bc == cd & cd == da & da == ab & ac != bd) {  
 mas\_con[1][i] = 3;mas\_con[2][i]= ((ac\*bd)/2);mas\_con[3][i]=-1;}  
 else if (ab == cd & bc == da & ac == bd) {  
 mas\_con[1][i] = 1; mas\_con[2][i]= (ab\*bc);mas\_con[3][i]=1;}  
 else if ((int) (*pow*(ac, 2) + *pow*(bd, 2)+1)== (int) (*pow*(ab, 2) + *pow*(cd, 2) + 2 \* bc \* da)) {  
 mas\_con[1][i] = 4;mas\_con[2][i] = (((bc+da)/2)\**sqrt*(*pow*(ab,2)-*pow*((*pow*(bc-da,2)+*pow*(ab,2)-*pow*(cd,2))/(2\*(bc-da)),2)));//2-it's square через 4 стороны трапеции  
 if (ab==cd){mas\_con[3][i]=1;}  
 else mas\_con[3][i]=-1;  
 }  
 else {mas\_con[1][i]=5;mas\_con[2][i]= ((ab\*da\*sinBAD+bc\*cd\*sinBCD)/2);if (m+z==180){  
 mas\_con[3][i]=1;}  
 else mas\_con[3][i]=-1;}  
  
 }  
 *printMas2*(mas\_con);  
 }  
 public static void printMas2(double[][] array) {  
 for (int i=0; i<=array.length-1; i++) {  
 for (int j=0; j<=array[0].length-1; j++) {  
 *out*.print(array[i][j]+"\t");  
 }  
 *out*.println();  
 }  
 }  
}

## 11.3. Тестирование работы программы с проверкой

# Задание 12

## 12.1. Постановка задачи

Дан двумерный квадратный массив коэффициентов системы линейных алгебраических уравнений A, одномерный массив столбца правой части СЛАУ F и одномерный массив столбца полученного решения X. При этом пользователь вводит размерность массива и данные сам и может допустить ошибку при вычислении решения СЛАУ или при вводе. Поэтому выполнить проверку соответствует ли полученное решение данной СЛАУ 

## 12.2. Решение задачи, код программы

import java.util.Arrays;  
import java.util.Scanner;  
  
import static java.lang.System.*out*;  
  
public class Zad12 {  
 public static void main(String[] args) {  
 Scanner mc =new Scanner(System.*in*);  
 System.*out*.println("Введите N");  
 int n = mc.nextInt();  
 int[][] mas\_1=new int[n][n];  
 int[] masF =new int[n];  
 int[] masX =new int[n];  
 int[] masSR=new int[n];  
 System.*out*.println("Заполните коэффициенты, заполняйте так- 1 строка, все столбцы, 2 строка и т.д.");  
 int j,i;  
 for (i=0;i<n;i++){  
 for(j=0;j<n;j++){  
 mas\_1[i][j]= mc.nextInt();  
 }  
 }  
 *printMas2*(mas\_1);  
 System.*out*.println("Заполните F и X");  
 for (i=0;i<n;i++){  
 System.*out*.print("F= ");  
 *out*.print("");masF[i]= mc.nextInt();  
 System.*out*.print("X= ");  
 *out*.print("");masX[i]= mc.nextInt();  
 }  
 for (i=0;i<n;i++){  
 for (j=0;j<n;j++){  
 masSR[i]+=mas\_1[i][j]\*masX[j];  
 }  
 }  
 if (Arrays.*equals*(masF,masSR)){  
 System.*out*.println("СЛАУ задано верно");  
 }  
 else System.*out*.println("СЛАУ задано не верно");  
 }  
 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++) {  
 *out*.print(array[i][j]+"\t");  
 }  
 *out*.println();  
 }  
 }  
}

## 12.3. Тестирование работы программы с проверкой

Для проверки задачи в MS Excel создана таблица данных в которой в диапазоне А2:B3 коэффициенты CЛАУ, а в диапазоне D2:D3 записаны значения столбца правой части СЛАУ F, в диапазоне F2:F3 значения решения XY. В диапазоне I2:I3 находится проверка введенных пользователем значения X. В диапазоне A5:A6 записана проверка на введенные значения F.

**Формулы:**

I2 =ЕСЛИ(D2=A5;1;0)

I3 =ЕСЛИ(D3=A6;1;0)

A5 =МУМНОЖ(A2:A2;F2:F3)

A6 = МУМНОЖ(B2:B3;F2:F3)

Далее в таблице 1 представлено тестирование работы программы с проверкой решения задачи на языке Java с решением задачи в MS Excel.

Тестирование работы программы и проверка результатов решения

|  |  |  |
| --- | --- | --- |
| № п.п. | Решение Java | Решение MS Excel |
| 1 |  | Снимок экрана 2023-03-22 225656 |
| 2 |  |  |

Сравнение решения задачи с использованием двух прикладных пакетов показала, что решения задачи в Java и MS Excel совпадает. Данный факт подтверждает правильность написанного кода программы. Неопределенностей при решении задачи выявлено не было, возможно данная ситуация связанна с малым количеством проверок данных.