Отчет по лабораторной работе №6

“Интегрирование и многопоточное программировние"

Выполнила: Качкуркина Арина Валерьевна

Группа: 6204-010302D

Год: 2025

Задание 1

Метод integrate(): я реализовала вычисление интеграла методом трапеций. Метод проверяет корректность параметров: границы должны быть в области определения функции, шаг > 0, левая граница < правой.

```
package threads;

import functions.Functions;

public class Integrator extends Thread {
    private Task task;
    private Semaphore semaphore;

    public Integrator(Task task, Semaphore semaphore) {
        this.task = task;
        this.semaphore = semaphore;
    }

    @Override
    public void run() {
        try {
            for (int i = 0; i < task.getNumberTask(); i++) {
                //проверка прерывания
                if (isInterrupted()) {
                    throw new InterruptedException("Integrator прерван");
                }

                //начинаем чтение (ждем, пока генератор запишет данные)
                semaphore.beginRead();

                //проверяем, что функция задана
                if (task.getFunction() != null) {
                    double left = task.getLeft();
                    double right = task.getRight();
                    double step = task.getStep();

                    try {
                        double result = Functions.integrate(task.getFunction(), left, right, step);
                        System.out.printf("Result %.6f %.6f %.6f %.6f%n", left, right, step, result);
                    } catch (IllegalArgumentException e) {
                        System.out.printf("Result %.6f %.6f %.6f ERROR: %s%n", left, right, step, e.getMessage());
                    }
                }

                //заканчиваем чтение (теперь можно писать)
                semaphore.endRead();


                try {
                    Thread.sleep(15);
                } catch (InterruptedException e) {
                    break;
                }
            }
        } catch (InterruptedException e) {
            System.out.println("Integrator прерван: " + e.getMessage());
        }
    }
}
```

Задание 2

Класс Task: создала для хранения параметров задания (функция, границы, шаг, количество заданий).

```
package threads;

import functions.Function;

public class Task {
    private Function function; //ссылка на объект интег. ф-ии
    private double left;       //левая граница
    private double right;      //правая граница
    private double step;       //шаг
    private int numberTask;    //поле, хранящее кол-во выполняемых заданий

    //ГЕТТЕРЫ
    //возвр. ссылку на объект ф-ии
    public Function getFunction() {
        return function;
    }
    //возвр. знач. левой границы
    public double getLeft() {
        return left;
    }
    //возвр. знач. правой границы
    public double getRight() {
        return right;
    }
    //возвр. знач. шага
    public double getStep() {
        return step;
    }
    //возвр. кол-во заданий, которые надо выполнить
    public int getNumberTask() {
        return numberTask;
    }

    //СЕТТЕРЫ

    public void setFunction(Function function) {
        this.function = function;
    }

    public void setLeft(double left) {
        this.left = left;
    }

    public void setRight(double right) {
        this.right = right;
    }

    public void setStep(double step) {
        this.step = step;
    }

    public void setNumberTask(int numberTask) {
        this.numberTask = numberTask;
    }
}
```

Задание 3

SimpleGenerator и SimpleIntegrator: два потока, работающие с общим объектом Task. Generator генерирует задания, Integrator их вычисляет.

Возникла проблема NullPointerException и несогласованные данные. Решение: добавила synchronized блоки при чтении/записи и проверку task.getFunction() != null.

```
package threads;

import functions.Functions;

public class SimpleIntegrator implements Runnable {
    private Task task;  //ссылка на объект задания

    //Конструктор (получает и сохраняет ссылку на Task)
    public SimpleIntegrator(Task task) {
        this.task = task;
    }

    @Override
    public void run() {
        //выполняем цикл столько же раз, сколько заданий нужно обраб.
        for (int i = 0; i < task.getNumberTask(); i++) {
            //блокируем объект task на время чтения параметров
            synchronized (task) {
                //проверка, что ф-ия уже задана (избавляемся от NullPointerException)
                if (task.getFunction() != null) { //есть ф-ия для вычисления
                    double left = task.getLeft();
                    double right = task.getRight();
                    double step = task.getStep();

                    try {
                        double result = Functions.integrate(task.getFunction(), left, right, step);
                        System.out.printf("Result %.6f %.6f %.6f %.6f%n", left, right, step, result);

                    } catch (IllegalArgumentException e) {
                        System.out.printf("Result %.6f %.6f %.6f ERROR: %s%n", left, right, step, e.getMessage());
                    }
                }

            }


            try {
                Thread.sleep(15);
            } catch (InterruptedException e) {
                break;
            }
        }
    }
}
```

```java
package threads;

import functions.Function;
import functions.basic.Log;
import java.util.Random;

public class SimpleGenerator implements Runnable {
    private Task task;  //ссылка на объект задания

    //Конструктор (получает и сохраняет ссылку на Task)
    public SimpleGenerator(Task task) {
        this.task = task;
    }

    @Override
    public void run() {
        Random random = new Random();

        //выполняем цикл столько же раз, сколько заданий нужно сгенерировать
        for (int i = 0; i < task.getNumberTask(); i++) {
            //блок. объект task на время генерации задания
            synchronized (task) {
                //создаем лог. ф-ию с ранд. основанием (от 1 до 10)
                double base = 1 + random.nextDouble() * 9;
                Log logFunc = new Log(base);

                //генерируем случайные параметры
                double left = random.nextDouble() * 100;
                double right = 100 + random.nextDouble() * 100;
                double step = random.nextDouble();

                //устанавливаем параметры в объект задания
                task.setFunction(logFunc);
                task.setLeft(left);
                task.setRight(right);
                task.setStep(step);

                //выводим сообщение о сгенерированных параметрах
                System.out.printf("Source %.6f %.6f %.6f%n", left, right, step);
            }

            try {
                Thread.sleep(15);
            } catch (InterruptedException e) {
                break;
            }
        }
    }
}
```

Задание 4

Улучшенная версия с семафорами. Я создала операций чтения/записи.

```java
package threads;

import functions.Function;
import functions.basic.Log;
import java.util.Random;

public class Generator extends Thread {
    private Task task;           //ссылка на объект
    private Semaphore semaphore; //ссылка на объект семафора

    //Конструктор получает и сохраняет ссылки
    public Generator(Task task, Semaphore semaphore) {
        this.task = task;
        this.semaphore = semaphore;
    }

    @Override
    public void run() {
        Random random = new Random();

        try {
            //выполняем цикл столько раз, сколько заданий нужно сгенерировать
            for (int i = 0; i < task.getNumberTask(); i++) {
                //проверяемработает ли поток
                if (isInterrupted()) {
                    throw new InterruptedException("Generator прерван");
                }

                //начинаем операцию записи (ждем, если интегратор еще читает)
                semaphore.beginWrite();

                //создаем лог. ф-ию со случ. основанием(от 1 до 10)
                double base = 1 + random.nextDouble() * 9;
                Log logFunc = new Log(base);

                //генерируем случ. параметры
                double left = random.nextDouble() * 100;
                double right = 100 + random.nextDouble() * 100;
                double step = random.nextDouble();

                //записываем параметры
                task.setFunction(logFunc);
                task.setLeft(left);
                task.setRight(right);
                task.setStep(step);

                //выводим сообщение о сгенерированных параметрах
                System.out.printf("Source %.6f %.6f %.6f%n", left, right, step);


                semaphore.endWrite();//заканчиваем


                try {
                    Thread.sleep(15);
                } catch (InterruptedException e) {
                    break;
                }
            }
        } catch (InterruptedException e) {
            //поток был прерван во время работы семафора
            System.out.println("Generator прерван: " + e.getMessage());
        }
    }
}
```

```java
package threads;

import functions.Functions;

public class Integrator extends Thread {
    private Task task;
    private Semaphore semaphore;

    public Integrator(Task task, Semaphore semaphore) {
        this.task = task;
        this.semaphore = semaphore;
    }

    @Override
    public void run() {
        try {
            for (int i = 0; i < task.getNumberTask(); i++) {
                //проверка прерывания
                if (isInterrupted()) {
                    throw new InterruptedException("Integrator прерван");
                }

                //начинаем чтение (ждем, пока генератор запишет данные)
                semaphore.beginRead();

                //проверяем, что функция задана
                if (task.getFunction() != null) {
                    double left = task.getLeft();
                    double right = task.getRight();
                    double step = task.getStep();

                    try {
                        double result = Functions.integrate(task.getFunction(), left, right, step);
                        System.out.printf("Result %.6f %.6f %.6f %.6f%n", left, right, step, result);
                    } catch (IllegalArgumentException e) {
                        System.out.printf("Result %.6f %.6f %.6f ERROR: %s%n", left, right, step, e.getMessage());
                    }
                }

                //заканчиваем чтение (теперь можно писать)
                semaphore.endRead();


                try {
                    Thread.sleep(15);
                } catch (InterruptedException e) {
                    break;
                }
            }
        } catch (InterruptedException e) {
            System.out.println("Integrator прерван: " + e.getMessage());
        }
    }
}
```

Все реализации работают корректно. Многопоточность ускоряет обработку, но требует синхронизации.

Результат работы программы:

ЗАДАНИЕ 1

Интеграл e^x от 0 до 1:

При шаге 0.001: 1.7182819716491948

Теоретическое: 1.718281828459045

Разница: 1.4319014973729338E-7

Поиск шага для точности 1e-7:

Найден шаг: 4.8828125E-4

Значение: 1.7182818625982395

Отклонение: 3.4139194360349734E-8

Делений шага: 12

Проверка точности:

Точность 1e-7 достигнута

ЗАДАНИЕ 2

Source 35.80142122263924 121.01666609634539 0.06608412700140298

Result 35.80142122263924 121.01666609634539 0.06608412700140298 13950.586872231002

Source 89.16156752549738 108.36399886345768 0.1143562571152249

Result 89.16156752549738 108.36399886345768 0.1143562571152249 45.896714282141986

Source 55.93808725785417 131.58223117894528 0.16519820257046236

Result 55.93808725785417 131.58223117894528 0.16519820257046236 150.81816284342122

Source 30.415935211022116 135.78329484322808 0.9073603772096667

Result 30.415935211022116 135.78329484322808 0.9073603772096667 201.3351532458923

Source 75.43695756621082 162.96635656905698 0.486447566205516

Result 75.43695756621082 162.96635656905698 0.486447566205516 200.98868808138667

Source 84.30032527747562 135.4082644999167 0.548003489471531

Result 84.30032527747562 135.4082644999167 0.548003489471531 204.13472564703275

Source 6.971971404793543 122.60513444169236 0.9179637529936467

Result 6.971971404793543 122.60513444169236 0.9179637529936467 878.357332115523

Source 14.287776730280177 161.47022742593555 0.5640620571047511

Result 14.287776730280177 161.47022742593555 0.5640620571047511 466.3205876650791

Source 93.46404679649474 110.52854050915082 0.5686890994708728

Result 93.46404679649474 110.52854050915082 0.5686890994708728 37.85341802770648

Source 90.98282557286652 158.00855661721886 0.4618676265643351

Result 90.98282557286652 158.00855661721886 0.4618676265643351 150.99168337197457

Source 44.79841779501926 171.0168744661664 0.41242062517168276

Result 44.79841779501926 171.0168744661664 0.41242062517168276 254.1346341849999

Source 76.81549827980761 168.39150717421853 0.45607465707464623

Result 76.81549827980761 168.39150717421853 0.45607465707464623 238.56849531221366

Source 48.71795943404571 123.36486171375878 0.15115333857913238

Result 48.71795943404571 123.36486171375878 0.15115333857913238 154.8915861332225

Source 19.48167352505863 153.82750771341628 0.7956807965497881

Result 19.48167352505863 153.82750771341628 0.7956807965497881 380.0137534598399

Source 81.28319789311105 137.7690487483401 0.6141289416293946

Result 81.28319789311105 137.7690487483401 0.6141289416293946 209.61078680415136

Source 92.02036310769583 199.38255227757628 0.12770569169134294

Result 92.02036310769583 199.38255227757628 0.12770569169134294 298.9949974414187

Source 67.28957712544373 147.35411007399787 0.25899655139988353

Result 67.28957712544373 147.35411007399787 0.25899655139988353 236.08635926973233

Source 42.418627302849956 111.01933835607035 0.5899349304034834

Result 42.418627302849956 111.01933835607035 0.5899349304034834 212.17184896729174

Source 66.03262360095681 112.35370125967985 0.5914913460796186

Result 66.03262360095681 112.35370125967985 0.5914913460796186 115.1232859536832

Source 40.6346614241235 132.17374335495757 0.5741352117512877

Result 40.6346614241235 132.17374335495757 0.5741352117512877 221.44274935237146

Source 82.81165417037461 177.46147525414347 0.5700375675864324

Result 82.81165417037461 177.46147525414347 0.5700375675864324 455.0369366870744

Source 35.92975466908439 133.3842784226285 0.351146603994797

Result 35.92975466908439 133.3842784226285 0.351146603994797 198.85566723123526

Source 31.562614314465808 188.36789136498922 0.311682078502334

Result 31.562614314465808 188.36789136498922 0.311682078502334 326.7014774758718

Source 62.39968858150612 153.49016486981753 0.9992590066541907

Result 62.39968858150612 153.49016486981753 0.9992590066541907 205.255204634017

Source 99.01070147011123 158.4984407201627 0.6657647790317596

Result 99.01070147011123 158.4984407201627 0.6657647790317596 177.12298402438748

Source 55.59623075340476 195.91360428951003 0.32727810344390884

Result 55.59623075340476 195.91360428951003 0.32727810344390884 303.4099009873461

Source 85.73338826432769 101.96890644237429 0.4717320593022828

Result 85.73338826432769 101.96890644237429 0.4717320593022828 34.884576769368145

Source 68.12934399966565 108.54194945536746 0.7336878102723897

Result 68.12934399966565 108.54194945536746 0.7336878102723897 87.1320155872033

Source 24.52447576105994 173.77329502080812 0.5674995256572878

Result 24.52447576105994 173.77329502080812 0.5674995256572878 328.50573786975855

Source 97.14073310069705 100.72530761611944 0.19331494472950828

Result 97.14073310069705 100.72530761611944 0.19331494472950828 17.752357425263433

Source 1.449783356223544 183.2200667963239 0.8647704072281778

Result 1.449783356223544 183.2200667963239 0.8647704072281778 562.7327647573774

Source 61.53935158107808 175.90527506890714 0.8129028008834145

Result 61.53935158107808 175.90527506890714 0.8129028008834145 248.94209219798913

Source 29.98512852261056 171.63226553667823 0.05771563556646353

Result 29.98512852261056 171.63226553667823 0.05771563556646353 357.89428552972015

Source 91.88949434492844 183.390391998498 0.6469994526664093

Result 91.88949434492844 183.390391998498 0.6469994526664093 370.49064828143656

Source 53.62719782473445 170.3271996048856 0.7814221153820116

Result 53.62719782473445 170.3271996048856 0.7814221153820116 285.5014363653289

Source 20.842108321721696 117.79395462553701 0.7155674026163292

Result 20.842108321721696 117.79395462553701 0.7155674026163292 210.70490329156067

Source 77.36794824912283 179.8912858731156 0.2865152760253443

Result 77.36794824912283 179.8912858731156 0.2865152760253443 216.60821657631038

Source 86.99994619821211 119.64979376732224 0.41472063022107486

Result 86.99994619821211 119.64979376732224 0.41472063022107486 774.2413657251014

Source 53.42315591635393 136.76569174891193 0.9463620715829857

Result 53.42315591635393 136.76569174891193 0.9463620715829857 182.30563777278874

Source 90.10716335269994 155.7216784609532 0.7619952766531474

Result 90.10716335269994 155.7216784609532 0.7619952766531474 141.85643473331115

Source 71.6057445148228 158.15838423095548 0.16326101841264307

Result 71.6057445148228 158.15838423095548 0.16326101841264307 218.1211648141174

Source 60.4540240454982 182.52192320495988 0.16756037031911353

Result 60.4540240454982 182.52192320495988 0.16756037031911353 515.9185322509528

Source 29.73450381701046 181.29821316551798 0.4900251474145749

Result 29.73450381701046 181.29821316551798 0.4900251474145749 320.70228698582275

Source 30.387308786187738 121.5548686451316 0.6727746139440262

Result 30.387308786187738 121.5548686451316 0.6727746139440262 262.0878052828692

Source 84.46884924469363 138.04668902983502 0.033364228897939885

Result 84.46884924469363 138.04668902983502 0.033364228897939885 208.9061654694341

Source 20.81706136437511 199.80083139698894 0.8474957791767529

Result 20.81706136437511 199.80083139698894 0.8474957791767529 729.9808531573342

Source 7.960220177881205 130.8830161033774 0.4554727508355001

Result 7.960220177881205 130.8830161033774 0.4554727508355001 310.48519444848824

Source 80.61030546250085 133.95615930300494 0.7875486027763704

Result 80.61030546250085 133.95615930300494 0.7875486027763704 170.23497751910588

Source 0.7107553079281015 101.27807488086246 0.14022953897890134

Result 0.7107553079281015 101.27807488086246 0.14022953897890134 1837.4045473817598

Source 41.76233879003281 161.94500942370317 0.9037578272231228

Result 41.76233879003281 161.94500942370317 0.9037578272231228 4390.066424783573

Source 36.02986880664476 106.47291086269406 0.34041552197964253

Result 36.02986880664476 106.47291086269406 0.34041552197964253 182.13598484388933

Source 78.23624285323025 150.21849921684162 0.4439522723635225

Result 78.23624285323025 150.21849921684162 0.4439522723635225 237.16800927077833

Source 36.47935890633026 145.6774213414984 0.37235577585711654

Result 36.47935890633026 145.6774213414984 0.37235577585711654 562.7857754071017

Source 81.36123641638142 130.35680424538415 0.9393463499650936

Result 81.36123641638142 130.35680424538415 0.9393463499650936 187.97488694526413

Source 7.697655526212166 101.70481743701212 0.17439220939780453

Result 7.697655526212166 101.70481743701212 0.17439220939780453 635.2257576665569

Source 89.68049065213222 143.48989235654645 0.016705451682717642

Result 89.68049065213222 143.48989235654645 0.016705451682717642 134.45578559436433

Source 79.99385269632398 189.0575000988681 0.7955929875741022

Result 79.99385269632398 189.0575000988681 0.7955929875741022 486.7873723312965

Source 36.139024351054395 196.05246096709487 0.26770101881505726

Result 36.139024351054395 196.05246096709487 0.26770101881505726 356.6838380778218

Source 33.10213017912933 124.47998618912126 0.6311050173139575

Result 33.10213017912933 124.47998618912126 0.6311050173139575 280.7308859162002

Source 0.5023202848703834 155.38874579050204 0.11686202982681815

Result 0.5023202848703834 155.38874579050204 0.11686202982681815 318.2490519906637

Source 4.487219725678415 160.99448983644743 0.4705557938754741

Result 4.487219725678415 160.99448983644743 0.4705557938754741 818.5730404814472

Source 79.89627289784458 138.72279818851754 0.8795288406751971

Result 79.89627289784458 138.72279818851754 0.8795288406751971 122.07655964547746

Source 6.59149726980599 151.43480932996133 0.6759108651024345

Result 6.59149726980599 151.43480932996133 0.6759108651024345 312.02077735789516

Source 15.359110655932728 174.00985850456436 0.4512155700238929

Result 15.359110655932728 174.00985850456436 0.4512155700238929 334.0503415583154

Source 3.4143706532596907 132.4941120785988 0.055192951044576666

Result 3.4143706532596907 132.4941120785988 0.055192951044576666 315.8965093385376

Source 37.2958875819794 136.18448887168728 0.31978777649631207

Result 37.2958875819794 136.18448887168728 0.31978777649631207 253.30637028579977

Source 21.81455834137096 145.30227907988058 0.24846929659354466

Result 21.81455834137096 145.30227907988058 0.24846929659354466 1536.4827056154722

Source 17.72620742312332 187.3210205818312 0.8418609420485129

Result 17.72620742312332 187.3210205818312 0.8418609420485129 800.060573567713

Source 97.92589519454467 133.5483947184807 0.6925132126619573

Result 97.92589519454467 133.5483947184807 0.6925132126619573 98.49544215677805

Source 42.21577125340984 174.22199962574382 0.9748266650757117

Result 42.21577125340984 174.22199962574382 0.9748266650757117 395.85799295888137

Source 84.0772838519797 185.16405451428665 0.4805462866989716

Result 84.0772838519797 185.16405451428665 0.4805462866989716 358.721236446313

Source 39.38239482143533 161.95024583998378 0.9059124275101821

Result 39.38239482143533 161.95024583998378 0.9059124275101821 300.425166194684

Source 66.60743896876744 108.58305950404005 0.6834673596308142

Result 66.60743896876744 108.58305950404005 0.6834673596308142 86.68183761494201

Source 30.867112523054573 106.82011464795372 0.26968129563198906

Result 30.867112523054573 106.82011464795372 0.26968129563198906 32138.961110618355

Source 69.9650048648192 159.5931736015349 0.9489745488539811

Result 69.9650048648192 159.5931736015349 0.9489745488539811 189.5590532121774

Source 89.66099121217621 197.36363718114384 0.8953779329404502

Result 89.66099121217621 197.36363718114384 0.8953779329404502 356.99844805973146

Source 47.4847184136858 159.7955342271454 0.44498660063409545

Result 47.4847184136858 159.7955342271454 0.44498660063409545 290.60231332277647

Source 19.678290846244938 116.78711659163403 0.275526860124669

Result 19.678290846244938 116.78711659163403 0.275526860124669 349.13644766560185

Source 84.2302223166361 172.687364984513 0.46376987388986346

Result 84.2302223166361 172.687364984513 0.46376987388986346 213.9275784010547

Source 94.46657506286779 105.43385410686491 0.19718958635019457

Result 94.46657506286779 105.43385410686491 0.19718958635019457 24.038773100995662

Source 62.29414024035488 128.38359036091907 0.25317705245060496

Result 62.29414024035488 128.38359036091907 0.25317705245060496 150.89400972890277

Source 17.94441986005426 126.7685744420887 0.22728658415305691

Result 17.94441986005426 126.7685744420887 0.22728658415305691 371.1312448620136

Source 63.94606292205246 112.99300793469513 0.8914039697231853

Result 63.94606292205246 112.99300793469513 0.8914039697231853 134.05129216386092

Source 95.05733167216617 165.39923082703217 0.9933904415855846

Result 95.05733167216617 165.39923082703217 0.9933904415855846 769.9027797597319

Source 13.019060445636722 163.6646098705588 0.2716706950108869

Result 13.019060445636722 163.6646098705588 0.2716706950108869 292.3561107521049

Source 25.2425028974794 110.65176745782072 0.4513965885883523

Result 25.2425028974794 110.65176745782072 0.4513965885883523 154.52869078190636

Source 79.92533477501115 135.31970546093703 0.5132333026729536

Result 79.92533477501115 135.31970546093703 0.5132333026729536 189.76285105871855

Source 56.59302883544042 104.21350398447767 0.58147752145073

Result 56.59302883544042 104.21350398447767 0.58147752145073 115.05776902240748

Source 0.6916642559944286 130.0854640613129 0.3651323318063119

Result 0.6916642559944286 130.0854640613129 0.3651323318063119 221.32538360617883

Source 63.612220795042326 155.36072804068553 0.20372709369773456

Result 63.612220795042326 155.36072804068553 0.20372709369773456 1055.4328363146271

Source 85.40522809589297 105.44548472911562 0.4190843799899454

Result 85.40522809589297 105.44548472911562 0.4190843799899454 41.459419868803494

Source 36.4486162356034 152.2861626612978 0.8549010133324959

Result 36.4486162356034 152.2861626612978 0.8549010133324959 249.71059442972535

Source 16.460473708093303 102.1230379912226 0.19792299523561918

Result 16.460473708093303 102.1230379912226 0.19792299523561918 234.87374556269566

Source 94.50234528828896 123.04190129981109 0.9176596900889402

Result 94.50234528828896 123.04190129981109 0.9176596900889402 62.89615718234335

Source 61.997624616459944 157.71261212662532 0.4359951681243983

Result 61.997624616459944 157.71261212662532 0.4359951681243983 321.68313146706447

Source 62.29344632402072 124.6134422735565 0.4437199079001315

Result 62.29344632402072 124.6134422735565 0.4437199079001315 209.60385295690926

Source 54.92347020022208 197.07535500008882 0.11791952121743288

Result 54.92347020022208 197.07535500008882 0.11791952121743288 821.9189812199065

Source 70.48277277977911 163.64045781712986 0.3623211905653929

Result 70.48277277977911 163.64045781712986 0.3623211905653929 222.30200970894

Source 12.90504764012198 115.51406394992499 0.09211278829309533

Result 12.90504764012198 115.51406394992499 0.09211278829309533 897.2520780267168

Source 69.93612339702014 177.29991988606525 0.015413715801257721

Result 69.93612339702014 177.29991988606525 0.015413715801257721 247.61902539387313

ЗАДАНИЕ 3

Source 64,636326 125,366157 0,171191

Result 64,636326 125,366157 0,171191 141,122797

Source 63,478448 188,507198 0,774409

Result 63,478448 188,507198 0,774409 580,529411

Source 28,277902 164,109052 0,429686

Result 28,277902 164,109052 0,429686 289,931379

Source 66,385483 113,054609 0,199802

Result 66,385483 113,054609 0,199802 106,272103

Source 78,444303 163,026681 0,277731

Result 78,444303 163,026681 0,277731 177,846447

Source 97,328034 197,665523 0,721866

Result 97,328034 197,665523 0,721866 306,488663

Source 53,668434 106,206176 0,978744

Result 53,668434 106,206176 0,978744 182,632344

Source 90,875644 131,269594 0,204067

Result 90,875644 131,269594 0,204067 275,971288

Source 47,183540 138,516657 0,229691

Result 47,183540 138,516657 0,229691 187,343444

Source 20,006446 146,034333 0,392444

Result 20,006446 146,034333 0,392444 236,048645

Source 28,772357 193,608423 0,340036

Result 28,772357 193,608423 0,340036 564,527264

Source 96,401730 102,447439 0,359978

Result 96,401730 102,447439 0,359978 42,801429

Source 76,729809 174,457353 0,739377

Result 76,729809 174,457353 0,739377 226,462053

Source 65,424396 106,827131 0,156585

Result 65,424396 106,827131 0,156585 82,157676

Source 45,329817 152,470055 0,130152

Result 45,329817 152,470055 0,130152 357,146179

Source 33,671818 139,776062 0,106145

Result 33,671818 139,776062 0,106145 287,890217

Source 18,186923 176,165455 0,851842

Result 18,186923 176,165455 0,851842 415,409442

Source 91,993844 199,094472 0,441971

Result 91,993844 199,094472 0,441971 272,351690

Source 22,903284 105,448424 0,273056

Result 22,903284 105,448424 0,273056 161,976366

Source 21,866481 185,668191 0,222125

Result 21,866481 185,668191 0,222125 401,560083

Source 7,391717 112,491068 0,277926

Result 7,391717 112,491068 0,277926 178,956066

Source 98,011918 185,847121 0,715981

Result 98,011918 185,847121 0,715981 281,056420

Source 3,265373 115,237304 0,163198

Result 3,265373 115,237304 0,163198 268,847244

Source 25,155528 128,829827 0,703454

Result 25,155528 128,829827 0,703454 336,059374

Source 26,684697 134,546364 0,512445

Result 26,684697 134,546364 0,512445 326,420193

Source 71,810441 122,569850 0,378293

Result 71,810441 122,569850 0,378293 253,125496

Source 92,052288 190,990828 0,863789

Result 92,052288 190,990828 0,863789 551,959207

Source 88,980342 196,719371 0,796836

Result 88,980342 196,719371 0,796836 719,423414

Source 38,946283 182,324838 0,237549

Result 38,946283 182,324838 0,237549 405,378707

Source 26,671099 106,125863 0,063655

Result 26,671099 106,125863 0,063655 165,876909

Source 97,682887 107,638078 0,840228

Result 97,682887 107,638078 0,840228 22,783932

Source 86,101857 173,397334 0,695479

Result 86,101857 173,397334 0,695479 190,390998

Source 43,898692 129,211201 0,511224

Result 43,898692 129,211201 0,511224 388,560253

Source 70,506224 174,051825 0,854805

Result 70,506224 174,051825 0,854805 230,102952

Source 17,543376 188,088274 0,728692

Result 17,543376 188,088274 0,728692 538,916148

Source 22,414156 192,766713 0,394605

Result 22,414156 192,766713 0,394605 869,902258

Source 6,673447 114,381891 0,240955

Result 6,673447 114,381891 0,240955 381,742786

Source 60,305204 104,243549 0,905543

Result 60,305204 104,243549 0,905543 87,784334

Source 21,437624 198,326597 0,639870

Result 21,437624 198,326597 0,639870 1300,610507

Source 52,523695 157,203956 0,207093

Result 52,523695 157,203956 0,207093 306,644071

Source 8,644503 113,096203 0,442552

Result 8,644503 113,096203 0,442552 282,270043

Source 91,346526 165,032277 0,973973

Result 91,346526 165,032277 0,973973 274,023886

Source 82,303109 105,400655 0,497870

Result 82,303109 105,400655 0,497870 851,536178

Source 55,094150 131,580578 0,439303

Result 55,094150 131,580578 0,439303 197,841438

Source 70,526999 194,878066 0,784117

Result 70,526999 194,878066 0,784117 353,603044

Source 71,175426 165,651607 0,365887

Result 71,175426 165,651607 0,365887 303,975654

Source 4,192482 148,297543 0,914262

Result 4,192482 148,297543 0,914262 514,235462

Source 17,956895 162,675390 0,727034

Result 17,956895 162,675390 0,727034 289,749831

Source 97,741645 119,588423 0,915183

Result 97,741645 119,588423 0,915183 44,762038

Source 28,347186 157,095029 0,788312

Result 28,347186 157,095029 0,788312 351,447166

Source 97,520758 171,711550 0,097299

Result 97,520758 171,711550 0,097299 182,581575

Source 82,987095 173,977361 0,531654

Result 82,987095 173,977361 0,531654 250,210079

Source 98,044723 121,388235 0,080998

Result 98,044723 121,388235 0,080998 51,817274

Source 5,936038 133,364100 0,613214

Result 5,936038 133,364100 0,613214 228,620399

Source 51,578144 172,702479 0,296275

Result 51,578144 172,702479 0,296275 258,114548

Source 0,745432 185,879084 0,296380

Result 0,745432 185,879084 0,296380 483,537630

Source 21,423715 130,592140 0,521973

Result 21,423715 130,592140 0,521973 708,460071

Source 89,809702 183,107237 0,051357

Result 89,809702 183,107237 0,051357 210,193774

Source 54,308466 110,244608 0,281053

Result 54,308466 110,244608 0,281053 266,009629

Source 44,885457 111,260901 0,762713

Result 44,885457 111,260901 0,762713 190,286296

Source 72,258561 183,412693 0,158280

Result 72,258561 183,412693 0,158280 347,863614

Source 21,669549 141,649135 0,736314

Result 21,669549 141,649135 0,736314 393,894186

Source 9,353243 127,666252 0,356743

Result 9,353243 127,666252 0,356743 421,030142

Source 36,326558 176,465622 0,947030

Result 36,326558 176,465622 0,947030 2360,294109

Source 43,748541 169,595270 0,059610

Result 43,748541 169,595270 0,059610 425,158878

Source 46,856388 193,410809 0,218643

Result 46,856388 193,410809 0,218643 528,907254

Source 0,361389 137,979969 0,413533

Result 0,361389 137,979969 0,413533 3404,977450

Source 27,192562 117,663886 0,277338

Result 27,192562 117,663886 0,277338 179,560615

Source 24,604770 167,943575 0,200088

Result 24,604770 167,943575 0,200088 284,665370

Source 46,921869 174,720266 0,870212

Result 46,921869 174,720266 0,870212 293,404124

Source 52,564511 122,103389 0,511687

Result 52,564511 122,103389 0,511687 151,344087

Source 94,515957 102,610143 0,390175

Result 94,515957 102,610143 0,390175 16,779542

Source 28,008340 119,735773 0,998339

Result 28,008340 119,735773 0,998339 214,534147

Source 6,965530 127,013801 0,698682

Result 6,965530 127,013801 0,698682 269,614055

Source 41,617867 189,970659 0,501688

Result 41,617867 189,970659 0,501688 395,905462

Source 93,707744 127,789259 0,087674

Result 93,707744 127,789259 0,087674 71,916522

Source 92,834068 190,031256 0,142717

Result 92,834068 190,031256 0,142717 336,838580

Source 67,737451 150,427907 0,572150

Result 67,737451 150,427907 0,572150 954,293459

Source 99,706932 154,534518 0,179543

Result 99,706932 154,534518 0,179543 752,395851

Source 21,792904 140,953744 0,954778

Result 21,792904 140,953744 0,954778 243,636519

Source 7,139878 166,868042 0,201687

Result 7,139878 166,868042 0,201687 970,840531

Source 12,506544 107,781832 0,353940

Result 12,506544 107,781832 0,353940 217,359269

Source 68,919580 168,659004 0,477246

Result 68,919580 168,659004 0,477246 229,901369

Source 40,411506 139,051618 0,136621

Result 40,411506 139,051618 0,136621 302,978061

Source 3,921806 148,350765 0,519361

Result 3,921806 148,350765 0,519361 1093,193664

Source 38,694460 128,950733 0,447560

Result 38,694460 128,950733 0,447560 184,835427

Source 51,883913 174,902768 0,188390

Result 51,883913 174,902768 0,188390 534,030047

Source 8,585419 191,544883 0,776050

Result 8,585419 191,544883 0,776050 1096,972803

Source 79,134626 167,629336 0,896194

Result 79,134626 167,629336 0,896194 192,753876

Source 64,275130 165,629484 0,251810

Result 64,275130 165,629484 0,251810 274,060758

Source 9,671696 118,193032 0,728942

Result 9,671696 118,193032 0,728942 195,392934

Source 75,250674 139,118637 0,611319

Result 75,250674 139,118637 0,611319 149,599290

Source 81,978109 105,768181 0,554836

Result 81,978109 105,768181 0,554836 67,269352

Source 18,939920 146,487087 0,026043

Result 18,939920 146,487087 0,026043 262,379758

Source 56,967426 171,390168 0,433172

Result 56,967426 171,390168 0,433172 483,094540

Source 81,555423 117,780794 0,233927

Result 81,555423 117,780794 0,233927 97,030975

Source 27,448906 142,506161 0,891136

Result 27,448906 142,506161 0,891136 2436,628860

Source 70,028427 182,711716 0,665591

Result 70,028427 182,711716 0,665591 247,899669

Source 4,503028 123,742885 0,884739

Result 4,503028 123,742885 0,884739 211,033435

Source 22,923555 158,352454 0,506619

Result 22,923555 158,352454 0,506619 1359,674869

ЗАДАНИЕ 4

Source 81,262697 138,462085 0,631458

Result 81,262697 138,462085 0,631458 142,173472

Source 81,106592 128,405529 0,720616

Result 81,106592 128,405529 0,720616 139,936384

Source 84,754844 153,142847 0,216754

Result 84,754844 153,142847 0,216754 1186,762238