Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет имени академика С.П. Королева»

Институт информатики и кибернетики

Кафедра технической кибернетики

Отчет по лабораторной работе №6

Дисциплина: «ООП»

Тема «Разработать приложение, формирующее в одном потоке вычислений набор заданий для интегрирования, а во втором потоке – вычисляющее значения интегралов функций.»

Выполнил: Зорин Дмитрий Сергеевич

Группа: 6201-120303D

Самара, 2025

Задание на лабораторную работу

Задание 1

В класс Functions добавил метод для вычисления интеграла функции методом трапеций

```
public static double integrate(Function f, double leftX, double rightX, double step) {  no usages
    if (leftX < f.getLeftDomainBorder() || rightX > f.getRightDomainBorder()) {
        throw new IllegalArgumentException("Интеграл выходит за область определения функции");
    }
    if (step <= 0) throw new IllegalArgumentException("Шаг должен быть > 0");

    double sum = 0;
    double x = leftX;

    while (x < rightX) {
        double nextX = Math.min(x + step, rightX);
        sum += (f.getFunctionValue(x) + f.getFunctionValue(nextX)) / 2 * (nextX - x);
        x = nextX;
    }

    return sum;
}
```

Метод делит интервал и суммирует площади трапеций

Задание 2

Во 2 задании я создал пакет Threads и описал класс Task:

```java
package functions.threads;

import functions.Function;

public class Task {  7 usages

    private Function function;  2 usages
    private double leftBorder;  2 usages
    private double rightBorder;  2 usages
    private double step;  2 usages
    private int tasksCount;  2 usages
    private boolean ready = false;  2 usages

    public synchronized boolean isReady() { return ready; }  1 usage
    public synchronized void setReady(boolean ready) { this.ready = ready; }  2 usages

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

    public double getLeftBorder() { return leftBorder; }  1 usage
    public void setLeftBorder(double leftBorder) { this.leftBorder = leftBorder; }  2 usages

    public double getRightBorder() { return rightBorder; }  1 usage
    public void setRightBorder(double rightBorder) { this.rightBorder = rightBorder; }  2 usages

    public double getStep() { return step; }  1 usage
    public void setStep(double step) { this.step = step; }  2 usages

    public int getTasksCount() { return tasksCount; }  3 usages
    public void setTasksCount(int tasksCount) { this.tasksCount = tasksCount; }  1 usage
}
```

А также метод nonThread():

```java
public static void nonThread() {  1 usage
    Task task = new Task();
    task.setTasksCount(100);
    Random rand = new Random();

    for (int i = 0; i < task.getTasksCount(); i++) {
        double base = 1 + 9 * rand.nextDouble();
        double left = 100 * rand.nextDouble();
        double right = 100 + 100 * rand.nextDouble();
        double step = rand.nextDouble();
        task.setFunction(new Log(base));
        task.setLeftBorder(left);
        task.setRightBorder(right);
        task.setStep(step);

        System.out.printf("Source %.5f %.5f %.5f%n", left, right, step);

        double integral = integrate(task.getFunction(), left, right, step);

        System.out.printf("Result %.5f %.5f %.5f %.10f%n", left, right, step, integral);
    }
}
```

Метод nonThread() показывает как последовательная обработка через методы объекта Taskc, а класс Functions позволяет интегрировать функции без многопоточности

Задание 3

Необходимо реализовать два потока: генератор задач и интегратор

Класс SimpleGenerator:

```java
package functions.threads;

import functions.Functions;
import functions.basic.Log;

import java.util.Random;

public class SimpleGenerator implements Runnable {  1 usage

    private Task task;  9 usages
    private Random rnd = new Random();  4 usages

    public SimpleGenerator(Task task) {  5 usages
        this.task = task;
    }

    @Override
    public void run() {
        for (int i = 0; i < task.getTasksCount(); i++) {
            double base = 1 + 9 * rnd.nextDouble();
            double left = rnd.nextDouble() * 100;
            double right = 100 + rnd.nextDouble() * 100;
            double step = rnd.nextDouble();

            synchronized (task) {
                task.setFunction(new Log(base));
                task.setLeftBorder(left);
                task.setRightBorder(right);
                task.setStep(step);
                task.setReady(true);
                task.notifyAll();
            }

            System.out.printf("Source %.5f %.5f %.5f%n", left, right, step);

            try { Thread.sleep( millis: 10); } catch (InterruptedException ignored) {}
```

run() — метод, который вызывается при старте потока

Класс SimpleIntegrator:

```java
package functions.threads;

import functions.Functions;

public class SimpleIntegrator implements Runnable {  1 usage

    private Task task;  10 usages

    public SimpleIntegrator(Task task) {  5 usages
        this.task = task;
    }

    @Override
    public void run() {
        for (int i = 0; i < task.getTasksCount(); i++) {
            synchronized (task) {
                while (!task.isReady()) {
                    try { task.wait(); } catch (InterruptedException ignored) {}
                }

                double left = task.getLeftBorder();
                double right = task.getRightBorder();
                double step = task.getStep();
                double result = Functions.integrate(task.getFunction(), left, right, step);

                System.out.printf("Result %.5f %.5f %.5f %.10f%n", left, right, step, result);

                task.setReady(false);
            }
        }
```

Задание 4

В задании 4 необходимо было использовать семафор для предотвращения пропуска задач интегратором

Класс семафора:

```java
package functions.threads;

public class Semaphore {  6 usages
    private boolean available = true;  5 usages

    public synchronized void acquireWrite() throws InterruptedException {  1 usage
        while (!available) wait();
        available = false;
    }

    public synchronized void releaseWrite() {  1 usage
        available = true;
        notifyAll();
    }

    public synchronized void acquireRead() throws InterruptedException {  1 usage
        while (available) wait();
    }

    public synchronized void releaseRead() {  1 usage
        available = true;
        notifyAll();
    }
```

Семафор различает операции чтения и записи в защищаемый объект
Также необходимо создать 2 класса Generator и Integrator:

Примет 1 из классов:

```java
package functions.threads;

import functions.Function;
import functions.Functions;

public class Integrator extends Thread {  no usages
    private Task task;  6 usages
    private Semaphore semaphore;  3 usages

    public Integrator(Task task, Semaphore semaphore) {  4 usages
        this.task = task;
        this.semaphore = semaphore;
    }

    @Override
    public void run() {
        try {
            for (int i = 0; i < task.getTasksCount(); i++) {
                semaphore.acquireRead();

                Function f = task.getFunction();
                double left = task.getLeftBorder();
                double right = task.getRightBorder();
                double step = task.getStep();

                double result = Functions.integrate(f, left, right, step);

                System.out.println("Result " + left + " " + right + " " + step + " " + result);

                semaphore.releaseRead();
            }
        } catch (InterruptedException ignored) { }
    }
```

В Main надо создать метод complicatedThreads(), который создаёт и запускает два потока Generator и Integrator и проверяет их работу

```java
public static void complicatedThreads() {  no usages
    Task task = new Task();
    task.setTasksCount(100);

    Semaphore semaphore = new Semaphore();

    Generator generator = new Generator(task, semaphore);
    Integrator integrator = new Integrator(task, semaphore);

    generator.start();
    integrator.start();

    try {
        Thread.sleep( millis: 50);
        generator.interrupt();
        integrator.interrupt();

        generator.join();
        integrator.join();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
```

Пример работы Main:
Non-thread
Source 31,14320 112,53376 0,22179
Result 31,14320 112,53376 0,22179 200,1918769589
Source 8,63712 162,60912 0,74072
Result 8,63712 162,60912 0,74072 347,6414881698
Source 41,65808 137,34222 0,32913
Result 41,65808 137,34222 0,32913 305,5621036790
Source 55,11847 149,08880 0,09660
Result 55,11847 149,08880 0,09660 191,9966257912
Source 57,26536 183,60060 0,00530
Result 57,26536 183,60060 0,00530 556,5309665213
Source 49,97842 162,86267 0,17263
Result 49,97842 162,86267 0,17263 406,1949793235
Source 31,85904 196,16022 0,48912
Result 31,85904 196,16022 0,48912 515,1125077642
Source 12,81460 166,13082 0,59851
Result 12,81460 166,13082 0,59851 338,9653866322

Source 94,09634 148,21827 0,03002
Result 94,09634 148,21827 0,03002 161,5229926541
Source 6,12755 141,72933 0,54739
Result 6,12755 141,72933 0,54739 274,1003196377
Source 37,96329 198,60472 0,58472
Result 37,96329 198,60472 0,58472 377,5364493084
Source 36,85128 172,05115 0,97630
Result 36,85128 172,05115 0,97630 404,9920438175
Source 4,11788 171,34097 0,05551
Result 4,11788 171,34097 0,05551 330,9094989224
Source 64,48170 101,37147 0,59759
Result 64,48170 101,37147 0,59759 74,8690555331
Source 39,90146 185,98587 0,01145
Result 39,90146 185,98587 0,01145 328,8660187131
Source 16,19362 122,59324 0,64779
Result 16,19362 122,59324 0,64779 232,7704906077
Source 50,90900 192,28471 0,08397
Result 50,90900 192,28471 0,08397 336,6742602900
Source 34,33689 154,42855 0,77029
Result 34,33689 154,42855 0,77029 265,0410044921
Source 82,00871 187,63681 0,47827
Result 82,00871 187,63681 0,47827 302,6857747289
Source 6,70227 100,03136 0,57827
Result 6,70227 100,03136 0,57827 179,5059475923
Source 90,70186 120,89736 0,29189
Result 90,70186 120,89736 0,29189 64,9231732922
Source 54,94775 103,09732 0,71429
Result 54,94775 103,09732 0,71429 137,1561821310
Source 0,71327 111,02908 0,58187
Result 0,71327 111,02908 0,58187 677,6490026426
Source 81,57588 142,44619 0,47290
Result 81,57588 142,44619 0,47290 282,0911945364
Source 48,70312 134,11935 0,37191
Result 48,70312 134,11935 0,37191 249,0382630617
Source 0,85276 131,10264 0,30271
Result 0,85276 131,10264 0,30271 226,0874045438
Source 61,99074 157,27854 0,04101
Result 61,99074 157,27854 0,04101 199,4397648362
Source 4,05925 110,89089 0,75431
Result 4,05925 110,89089 0,75431 215,6472603578

Source 30,18101 122,93493 0,78478
Result 30,18101 122,93493 0,78478 175,7247876365
Source 34,46520 158,55894 0,69054
Result 34,46520 158,55894 0,69054 245,2168529419
Source 58,58390 112,04207 0,09115
Result 58,58390 112,04207 0,09115 257,6544603034
Source 72,41707 144,44793 0,82529
Result 72,41707 144,44793 0,82529 151,2338300231
Source 16,31796 139,71410 0,42152
Result 16,31796 139,71410 0,42152 252,7450176708
Source 55,92978 109,72230 0,90753
Result 55,92978 109,72230 0,90753 224,2905292782
Source 16,98817 161,80468 0,53912
Result 16,98817 161,80468 0,53912 325,0862709236
Source 87,17331 153,17641 0,11670
Result 87,17331 153,17641 0,11670 297,7152650855
Source 94,82892 195,94514 0,17568
Result 94,82892 195,94514 0,17568 324,5555608665
Source 0,24474 143,63312 0,48610
Result 0,24474 143,63312 0,48610 283,8520579763
Source 64,20272 185,55748 0,97801
Result 64,20272 185,55748 0,97801 391,3482218884
Source 90,61623 198,95011 0,42050
Result 90,61623 198,95011 0,42050 297,6143510766
Source 75,34524 160,35784 0,92583
Result 75,34524 160,35784 0,92583 327,4461586860
Source 31,64975 197,64947 0,62448
Result 31,64975 197,64947 0,62448 1333,0792776019
Source 4,24723 171,86610 0,66222
Result 4,24723 171,86610 0,66222 801,3546271169
Source 28,56900 114,18616 0,00792
Result 28,56900 114,18616 0,00792 165,5351872714
Source 39,56421 107,43720 0,42817
Result 39,56421 107,43720 0,42817 163,6944319383
Source 85,27930 193,87107 0,38495
Result 85,27930 193,87107 0,38495 451,0267142462
Source 54,44418 111,32979 0,54261
Result 54,44418 111,32979 0,54261 416,6797139158
Source 84,42595 108,46422 0,03286
Result 84,42595 108,46422 0,03286 58,3242242117

Source 76,16427 166,32791 0,61926
Result 76,16427 166,32791 0,61926 493,5104714821
Source 6,08179 158,45377 0,42732
Result 6,08179 158,45377 0,42732 442,3231895071
Source 91,46685 114,43119 0,96186
Result 91,46685 114,43119 0,96186 153,3095130491
Source 35,45447 113,03643 0,72036
Result 35,45447 113,03643 0,72036 1296,4997418670
Source 50,48571 120,12939 0,82897
Result 50,48571 120,12939 0,82897 328,8707096411
Source 35,00422 103,81488 0,77747
Result 35,00422 103,81488 0,77747 271,9722589010
Source 57,15293 104,77667 0,50998
Result 57,15293 104,77667 0,50998 167,5232889055
Source 81,36250 113,22792 0,84027
Result 81,36250 113,22792 0,84027 65,3677811183
Source 58,97715 105,58145 0,40354
Result 58,97715 105,58145 0,40354 409,1622410718
Source 92,44249 123,72887 0,13111
Result 92,44249 123,72887 0,13111 71,6437344499
Source 74,08374 137,72729 0,53995
Result 74,08374 137,72729 0,53995 262,5025868925
Source 95,13166 118,72791 0,96608
Result 95,13166 118,72791 0,96608 49,1790278327
Source 18,40756 163,14057 0,23884
Result 18,40756 163,14057 0,23884 364,7273635757
Source 8,95738 151,31391 0,19269
Result 8,95738 151,31391 0,19269 764,5541291640
Source 20,92904 194,42044 0,84355
Result 20,92904 194,42044 0,84355 524,3095657470
Source 46,11600 121,77151 0,01887
Result 46,11600 121,77151 0,01887 181,3258924095
Source 39,84391 187,44684 0,72372
Result 39,84391 187,44684 0,72372 1773,7058019786
Source 53,25608 129,90890 0,54942
Result 53,25608 129,90890 0,54942 157,2145506794
Source 17,55757 101,18205 0,98729
Result 17,55757 101,18205 0,98729 277,4377029060
Source 45,40640 193,21209 0,59650
Result 45,40640 193,21209 0,59650 371,2588013880

Source 37,39445 128,18150 0,59359
Result 37,39445 128,18150 0,59359 353,9557243399
Source 7,43940 146,84118 0,58051
Result 7,43940 146,84118 0,58051 323,3634959071
Source 69,91310 124,32945 0,80859
Result 69,91310 124,32945 0,80859 143,9003500513
Source 63,46056 166,48431 0,58781
Result 63,46056 166,48431 0,58781 551,8943108720
Source 21,71764 191,31510 0,19411
Result 21,71764 191,31510 0,19411 387,6115847918
Source 39,51457 180,08488 0,90154
Result 39,51457 180,08488 0,90154 368,2721018111
Source 24,65616 190,36486 0,57201
Result 24,65616 190,36486 0,57201 705,4333893966
Source 69,99863 159,45772 0,57215
Result 69,99863 159,45772 0,57215 374,1905903198
Source 28,29669 190,74630 0,65505
Result 28,29669 190,74630 0,65505 371,0687458987
Source 23,17373 107,80981 0,84394
Result 23,17373 107,80981 0,84394 238,9189990193
Source 83,57141 181,55949 0,28571
Result 83,57141 181,55949 0,28571 234,2490485451
Source 46,16445 184,18107 0,50394
Result 46,16445 184,18107 0,50394 1924,5409095558
Source 30,47969 100,33129 0,47431
Result 30,47969 100,33129 0,47431 152,9617606201
Source 54,83792 130,82110 0,46972
Result 54,83792 130,82110 0,46972 468,1418945666
Source 27,74059 156,92827 0,67804
Result 27,74059 156,92827 0,67804 257,9087967444
Source 18,95248 194,51717 0,18241
Result 18,95248 194,51717 0,18241 425,9602307243
Source 87,28543 156,30382 0,01207
Result 87,28543 156,30382 0,01207 163,8222496778
Source 17,96686 157,88937 0,62129
Result 17,96686 157,88937 0,62129 394,8431072507
Source 1,48112 148,59520 0,33431
Result 1,48112 148,59520 0,33431 283,6933648355
Source 67,90025 121,66839 0,00656
Result 67,90025 121,66839 0,00656 262,0375575286

Source 75,97479 160,08591 0,39515
Result 75,97479 160,08591 0,39515 452,6348595733
Source 44,63419 103,97155 0,50643
Result 44,63419 103,97155 0,50643 171,7938753877
Source 6,05036 183,54936 0,49365
Result 6,05036 183,54936 0,49365 382,8342114097
Source 54,04652 147,59283 0,51518
Result 54,04652 147,59283 0,51518 186,3230845960
Source 59,38778 131,48909 0,33146
Result 59,38778 131,48909 0,33146 369,7045338041
Source 24,36846 151,32235 0,31845
Result 24,36846 151,32235 0,31845 402,3699309878
Source 53,11540 180,79147 0,00609
Result 53,11540 180,79147 0,00609 267,7625358798
Source 21,15502 166,31108 0,21214
Result 21,15502 166,31108 0,21214 291,0164415849
Source 38,04621 162,30634 0,33467
Result 38,04621 162,30634 0,33467 433,6337732505
Source 10,25969 123,49632 0,39151
Result 10,25969 123,49632 0,39151 306,9440347390
Source 27,98858 173,87118 0,94983
Result 27,98858 173,87118 0,94983 366,8306704101
Source 19,20384 183,10358 0,17668
Result 19,20384 183,10358 0,17668 887,7909753589

Simple threads
Source 45,46919 159,55237 0,29180
Result 45,46919 159,55237 0,29180 269,7066919155
Source 39,04139 124,64815 0,49820
Result 39,04139 124,64815 0,49820 169,0617892949
Source 95,74655 106,49066 0,03294
Result 95,74655 106,49066 0,03294 24,1777519494
Source 37,09320 169,31413 0,36849
Result 37,09320 169,31413 0,36849 303,8947717271
Source 77,19034 159,70645 0,95877
Result 77,19034 159,70645 0,95877 267,4772838562
Source 95,53314 199,28772 0,80926
Result 95,53314 199,28772 0,80926 261,2127153651
Source 87,74666 135,48978 0,43858
Result 87,74666 135,48978 0,43858 116,2529646934

Source 73,09921 183,47192 0,62145
Result 73,09921 183,47192 0,62145 233,4252851324
Source 8,78850 117,30593 0,15576
Result 8,78850 117,30593 0,15576 378,6221656893
Source 85,31986 171,78584 0,78043
Result 85,31986 171,78584 0,78043 243,4052382239
Source 54,73753 197,56571 0,28537
Result 54,73753 197,56571 0,28537 432,3316830883
Source 74,03624 171,80139 0,80143
Result 74,03624 171,80139 0,80143 279,0183315184
Source 29,77184 104,06977 0,23585
Result 29,77184 104,06977 0,23585 883,1415220278
Source 46,70586 193,06172 0,82711
Result 46,70586 193,06172 0,82711 321,6561179099
Source 73,26071 183,39265 0,83359
Result 73,26071 183,39265 0,83359 236,0478704540
Source 79,62836 143,84720 0,44138
Result 79,62836 143,84720 0,44138 306,9151878147
Source 0,56665 138,47216 0,07934
Result 0,56665 138,47216 0,07934 275,5244418807
Source 4,23414 190,15461 0,46318
Result 4,23414 190,15461 0,46318 411,3239488589
Source 70,85554 142,66965 0,85540
Result 70,85554 142,66965 0,85540 530,7959560568
Source 20,57618 111,65495 0,22426
Result 20,57618 111,65495 0,22426 199,1128852506
Source 38,55552 177,56689 0,21242
Result 38,55552 177,56689 0,21242 415,6791083822
Source 7,16602 114,39577 0,08475
Result 7,16602 114,39577 0,08475 197,9492524431
Source 70,10509 166,76520 0,11479
Result 70,10509 166,76520 0,11479 292,8715033001
Source 21,29427 185,01346 0,19060
Result 21,29427 185,01346 0,19060 1331,6095576147
Source 32,63120 102,95228 0,77349
Result 32,63120 102,95228 0,77349 133,7172567579
Source 8,97637 196,22112 0,72655
Result 8,97637 196,22112 0,72655 931,4333363859
Source 86,19789 190,50261 0,94529
Result 86,19789 190,50261 0,94529 239,8723354746

Source 68,66425 198,33777 0,16876
Result 68,66425 198,33777 0,16876 638,3131967288
Source 36,26606 150,87681 0,82721
Result 36,26606 150,87681 0,82721 246,2046277490
Source 37,86446 156,62442 0,29093
Result 37,86446 156,62442 0,29093 320,8100071308
Source 28,00515 163,08582 0,42525
Result 28,00515 163,08582 0,42525 262,9019152000
Source 96,80344 119,19263 0,92772
Result 96,80344 119,19263 0,92772 435,9329128590
Source 89,09714 136,48529 0,52609
Result 89,09714 136,48529 0,52609 385,4098317925
Source 8,28048 125,25123 0,74957
Result 8,28048 125,25123 0,74957 396,3559704465
Source 52,67367 138,16598 0,65087
Result 52,67367 138,16598 0,65087 179,8362421823
Source 37,84915 188,45675 0,55264
Result 37,84915 188,45675 0,55264 347,6971257912
Source 17,02687 122,20574 0,61983
Result 17,02687 122,20574 0,61983 195,9422877131
Source 23,46761 115,13333 0,81786
Result 23,46761 115,13333 0,81786 167,9983084251
Source 23,73437 133,04861 0,60680
Result 23,73437 133,04861 0,60680 230,1718802373
Source 88,75863 142,12755 0,54303
Result 88,75863 142,12755 0,54303 135,7086052290
Source 60,83943 180,27685 0,81230
Result 60,83943 180,27685 0,81230 1266,4756943236
Source 19,01980 104,06392 0,60183
Result 19,01980 104,06392 0,60183 229,1018346814
Source 22,72642 107,04133 0,13443
Result 22,72642 107,04133 0,13443 178,5949951794
Source 44,12463 164,76664 0,13359
Result 44,12463 164,76664 0,13359 264,5306146791
Source 83,33267 199,41605 0,46611
Result 83,33267 199,41605 0,46611 711,2825892743
Source 97,45512 172,61327 0,71787
Result 97,45512 172,61327 0,71787 232,2467978499
Source 64,33700 108,94232 0,49788
Result 64,33700 108,94232 0,49788 210,4093889750

Result 45,00653 124,58585 0,33765 240,4049754339
Source 45,00653 124,58585 0,33765
Source 4,96574 185,62825 0,15494
Result 4,96574 185,62825 0,15494 547,1964506770
Source 3,80475 148,78519 0,38503
Result 3,80475 148,78519 0,38503 264,0720085357
Source 65,05675 143,66929 0,04755
Result 65,05675 143,66929 0,04755 231,2255215586
Source 93,77719 140,17394 0,21339
Result 93,77719 140,17394 0,21339 98,2312639306
Source 18,02223 107,15517 0,56344
Result 18,02223 107,15517 0,56344 166,7889840601
Source 92,21758 109,31397 0,98997
Result 92,21758 109,31397 0,98997 346,0051736926
Source 15,35071 108,02919 0,48142
Result 15,35071 108,02919 0,48142 179,5062390175
Source 84,98537 128,86387 0,85258
Result 84,98537 128,86387 0,85258 93,7121680493
Source 29,28907 133,56528 0,23483
Result 29,28907 133,56528 0,23483 210,0857913565
Source 33,58047 187,60199 0,62462
Result 33,58047 187,60199 0,62462 481,1677567210
Source 5,34504 145,62846 0,20394
Result 5,34504 145,62846 0,20394 316,2604213512
Source 2,23375 186,74078 0,59206
Result 2,23375 186,74078 0,59206 370,8013940321
Source 81,25096 118,98445 0,66613
Result 81,25096 118,98445 0,66613 223,8205186472
Source 14,15411 138,29802 0,09646
Result 14,15411 138,29802 0,09646 910,0801411694
Source 54,02925 170,92392 0,28265
Result 54,02925 170,92392 0,28265 287,3017952494
Source 31,07263 150,29053 0,94784
Result 31,07263 150,29053 0,94784 230,6153237426
Source 1,80043 179,46711 0,25110
Result 1,80043 179,46711 0,25110 422,6929370639
Source 75,22450 108,51606 0,91141
Result 75,22450 108,51606 0,91141 87,2712095415
Source 91,97408 112,41434 0,50233
Result 91,97408 112,41434 0,50233 48,3191626452

Source 37,90925 189,68888 0,77189
Result 37,90925 189,68888 0,77189 375,7071397202
Source 32,46359 114,50814 0,57075
Result 32,46359 114,50814 0,57075 173,5620336564
Source 43,77006 165,69483 0,72952
Result 43,77006 165,69483 0,72952 305,2548194709
Source 47,56509 124,97254 0,01697
Result 47,56509 124,97254 0,01697 191,0308335295
Source 64,93266 183,73130 0,13155
Result 64,93266 183,73130 0,13155 407,1872305432
Source 85,14145 122,69526 0,53160
Result 85,14145 122,69526 0,53160 86,8884712172
Source 4,14441 132,09792 0,79314
Result 4,14441 132,09792 0,79314 573,0674816243
Source 86,75172 121,31822 0,09926
Result 86,75172 121,31822 0,09926 90,7323491095
Source 91,05434 126,40342 0,41160
Result 91,05434 126,40342 0,41160 86,6917597810
Source 77,36685 161,50223 0,85152
Result 77,36685 161,50223 0,85152 464,0420542503
Source 60,77466 175,95450 0,82785
Result 60,77466 175,95450 0,82785 250,6221866671
Source 56,31212 172,63132 0,48782
Result 56,31212 172,63132 0,48782 488,5748595263
Source 8,86434 115,09206 0,13903
Result 8,86434 115,09206 0,13903 448,2371107717
Source 96,07709 101,22245 0,39324
Result 96,07709 101,22245 0,39324 47,7673139352
Source 41,87548 109,24072 0,91608
Result 41,87548 109,24072 0,91608 128,3820400937
Source 80,63352 128,30609 0,83201
Result 80,63352 128,30609 0,83201 118,1919444661
Source 67,60306 170,43695 0,03040
Result 67,60306 170,43695 0,03040 381,4784864841
Source 39,90208 123,04340 0,84560
Result 39,90208 123,04340 0,84560 302,8323299055
Source 94,17865 142,10170 0,84323
Result 94,17865 142,10170 0,84323 122,6060590867
Source 62,00826 104,84569 0,73449
Result 62,00826 104,84569 0,73449 92,5552659778

Source 32,14874 137,82300 0,76191
Result 32,14874 137,82300 0,76191 336,8867044020
Source 89,96826 111,26154 0,25231
Result 89,96826 111,26154 0,25231 62,7146631995
Source 18,67835 143,91416 0,41666
Result 18,67835 143,91416 0,41666 358,1831644827
Source 34,44543 193,89753 0,89137
Result 34,44543 193,89753 0,89137 407,9809041303
Source 8,02045 120,51329 0,96211
Result 8,02045 120,51329 0,96211 494,5722888145
Source 59,21794 189,63792 0,88850
Result 59,21794 189,63792 0,88850 301,1631680444
Source 97,39197 160,98965 0,25714
Result 97,39197 160,98965 0,25714 7290,6598019814
Source 20,99814 139,42912 0,83003
Result 20,99814 139,42912 0,83003 359,3848292811
Source 82,39681 194,42256 0,92724
Result 82,39681 194,42256 0,92724 317,9243723903
Source 70,24449 151,89440 0,98486
Result 70,24449 151,89440 0,98486 241,7984362312
Source 61,13666 140,91842 0,56308
Result 61,13666 140,91842 0,56308 319,5080614500
Source 44,60643 134,04464 0,74457
Result 44,60643 134,04464 0,74457 376,3214166415
Source 31,15024 154,53147 0,55928
Result 31,15024 154,53147 0,55928 328,7664959684