**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего образования «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Центр ускоренного обучения

Отчет по решению задач с сайта acm.timus

по дисциплине «Программирование»

Старший преподаватель: Н.А. Архипов

Студент гр. РИВ-120938у И.И. Корепанов

Екатеринбург 2022

2. Цель работы: Решить 20 любых задач с сайта <https://acm.timus.ru/>. Оформить отчет проделанной работы. Закрепить ссылку на github.

3. Описание задачи:

3.1. Решение 20 задач с сайта acm.timus;

3.2. Описание результата выполнения лабораторной работы.

4. Ход выполнения:

4.1. Задания для самостоятельной работы:  
Листинг 1. Код программы «Timus\_1000»

**/\***

**Задача #1000**

**Условие:**

**Вычислите a+b**

**\*/**

**import java.util.Scanner;**

**public class Timus\_1000**

**{**

**public static void main(String[] args)**

**{**

**Scanner in = new Scanner(System.in);**

**int a = in.nextInt();**

**int b = in.nextInt();**

**System.out.println(a + b);**

**}**

**}**

Листинг 2. Код программы «Timus\_1001»

**/\***

**Задача #1001**

**Исходные данные:**

**Входной поток содержит набор целых чисел Ai (0 ≤ Ai ≤ 1018), отделённых друг от друга**

**произвольным количеством пробелов и переводов строк. Размер входного потока не превышает 256 КБ.**

**Результат:**

**Для каждого числа Ai, начиная с последнего и заканчивая первым, в отдельной строке вывести его**

**квадратный корень не менее чем с четырьмя знаками после десятичной точки.**

**\*/**

**import java.io.\*;**

**import java.util.\*;**

**public class Timus\_1001**

**{**

**static BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));**

**public static void main(String[] args) throws IOException**

**{**

**StringTokenizer tokenizer;**

**Stack<Double> stack = new Stack<Double>();**

**String line;**

**while ( (line = reader.readLine()) != null )**

**{**

**tokenizer = new StringTokenizer(line);**

**while (tokenizer.hasMoreTokens())**

**{**

**stack.add(Math.sqrt(Double.parseDouble(tokenizer.nextToken())));**

**}**

**}**

**while (!stack.empty())**

**{**

**System.out.println(stack.pop());**

**}**

**}**

**}**

Листинг 3. Код программы «Timus\_1197»

**/\***

**Задача #1197**

**Исходные данные:**

**В первой строке находится единственное число N, 1 ≤ N ≤ 64 — количество тестов.**

**В каждой из последующих N строк содержится очередной тест: два символа**

**(маленькая латинская буква от 'a' до 'h' и цифра от 1 до 8) — стандартное шахматное**

**обозначение клетки, на которой стоит конь. При этом буква обозначает вертикаль,**

**а цифра — горизонталь.**

**Результат:**

**Выведите N строк: в каждой из них должно находиться единственное число — количество клеток**

**шахматной доски, находящихся под боем коня**

**\*/**

**import java.io.\*;**

**import java.util.\*;**

**public class Timus\_1197**

**{**

**static BufferedReader buffer = new BufferedReader(new InputStreamReader(System.in));**

**static PrintWriter out = new PrintWriter(System.out);**

**public static void main(String[] args) throws IOException**

**{**

**final int size = 8;**

**int dx[] = {-2, -2, 2, 2, -1, 1, -1, 1};**

**int dy[] = {1, -1, 1, -1, 2, 2, -2, -2};**

**int n = Integer.parseInt(buffer.readLine());**

**List<String> tests = new ArrayList<String>(n);**

**for (int i = 0; i < n; i++)**

**{**

**tests.add(buffer.readLine().toLowerCase());**

**}**

**for (String next : tests)**

**{**

**int column = next.charAt(0) - 'a';**

**int row = next.charAt(1) - '1';**

**int positionCount = 0;**

**int currentX = 0;**

**int currentY = 0;**

**for (int i = 0; i < size; i++)**

**{**

**currentX = column + dx[i];**

**currentY = row + dy[i];**

**if (currentX >= 0 && currentX < size && currentY >= 0 && currentY < size)**

**{**

**positionCount++;**

**}**

**}**

**out.println(positionCount);**

**}**

**out.flush();**

**}**

**}**

Листинг 4. Код программы «Timus\_1264»

**/\***

**Задача #1264**

**Исходные данные:**

**Единственная строка содержит целые числа N (0 ≤ N ≤ 40000) и M (0 ≤ M ≤ 40000).**

**Результат:**

**Выведите число секунд, требуемых Петечкину на написание функции.**

**\*/**

**import java.util.\*;**

**public class Timus\_1264**

**{**

**static Scanner in = new Scanner(System.in);**

**public static void main(String[] args)**

**{**

**int N = in.nextInt();**

**int M = in.nextInt() + 1;**

**int count = M \* N;**

**System.out.print(count);**

**}**

**}**

Листинг 5. Код программы «Timus\_1293»

**/\***

**Задача #1293**

**Исходные данные:**

**Единственная строка содержит целые числа N (1 ≤ N ≤ 100), A (1 ≤ A ≤ 100), B (1 ≤ B ≤ 100).**

**Результат:**

**Выведите вес необходимого для обработки сульфида тория в нанограммах.**

**\*/**

**import java.io.PrintWriter;**

**import java.util.Scanner;**

**public class Timus\_1293**

**{**

**public static void main(String[] args)**

**{**

**Scanner in = new Scanner(System.in);**

**PrintWriter out = new PrintWriter(System.out);**

**int N = in.nextInt();**

**int A = in.nextInt();**

**int B = in.nextInt();**

**if (N >= 1 && N <= 100 && B >= 1 && B <= 100 && A >= 1 && A <= 100)**

**{**

**out.println(A \* B \* N \* 2);**

**}**

**out.flush();**

**}**

**}**

Листинг 6. Код программы «Timus\_1297»

**/\***

**Задача #1297**

**Исходные данные:**

**Содержит всего одну строчку (не длиннее 1000 символов), содержащую только буквы латинского алфавита**

**(все пробелы и знаки препинания из строки Вася уже удалил).**

**Результат:**

**Максимальная по длине подстрока, читающаяся одинаково в обоих направлениях.**

**Если максимальных по длине подстрок больше одной, вывести самую левую из них.**

**\*/**

**import java.util.Scanner;**

**public class Timus\_1297**

**{**

**public static void main(String[] args)**

**{**

**Scanner sc = new Scanner(System.in);**

**String S = sc.nextLine();**

**int N = S.length();**

**int[][] T = new int[N][N];**

**int st = 0;**

**int end = 0;**

**int max = 0;**

**for (int k = 0; k < N; k++)**

**{**

**for (int h = k; h < N; h++)**

**{**

**int i = h - k;**

**int j = h;**

**if (i == j) {**

**T[i][j] = 1;**

**} else if (S.charAt(i) == S.charAt(j))**

**{**

**if (i + 1 == j)**

**{**

**T[i][j] = 1;**

**} else{**

**T[i][j] = T[i + 1][j - 1];**

**}**

**if (j - i > max && T[i][j] == 1)**

**{**

**max = j - i;**

**st = i;**

**end = j;**

**}**

**} else**

**{**

**T[i][j] = 0;**

**}**

**}**

**}**

**System.out.println(S.substring(st, end + 1));**

**}**

**}**

Листинг 7. Код программы «Timus\_1349»

**/\***

**Задача #1349**

**Исходные данные:**

**n (0 ≤ n ≤ 100)**

**Результат:**

**Три различных целых числа (а, b, c) таких, что аn + bn = cn, 1 ≤ a, b, c ≤ 100. Если решений несколько, вывести то, где a минимально.**

**Если и таких несколько, вывести то, где минимально b и т.д. Вывести −1, если решения нет.**

**\*/**

**import java.util.Scanner;**

**public class Timus\_1349**

**{**

**public static void main(String[] args)**

**{**

**Scanner scann = new Scanner(System.in);**

**int numb = scann.nextInt();**

**for (int a = 1; a <= 98; a++)**

**{**

**for (int b = a + 1; b <= 99 ; b++)**

**{**

**for (int c = b + 1; c <= 100; c++)**

**{**

**if(Math.pow(a, numb) + Math.pow(b, numb) == Math.pow(c, numb))**

**{**

**System.out.println(a + " "+ b +" "+ c);**

**System.exit(0);**

**}**

**}**

**}**

**}**

**System.out.println("-1");**

**}**

**}**

Листинг 8. Код программы «Timus\_1409»

**/\***

**Задача #1409**

**Исходные данные:**

**В единственной строке записано 2 числа — количество банок, простреленных Гарри и Ларри соответственно.**

**Результат:**

**Выведите 2 числа — количество банок, не простреленных Гарри и Ларри соответственно.**

**\*/**

**import java.io.PrintWriter;**

**import java.util.\*;**

**public class Timus\_1409**

**{**

**public static void main(String[] args)**

**{**

**Scanner in = new Scanner(System.in);**

**int [] result = new int[2];**

**for (int i = 0; i <= result.length -1; i++)**

**{**

**result[i] = in.nextInt();**

**}**

**in.close();**

**int tottalTargets = 0;**

**for (int target: result)**

**{**

**tottalTargets = tottalTargets +target;**

**}**

**for (int can: result)**

**{**

**System.out.printf("%s \n", tottalTargets-can-1);**

**}**

**}**

**}**

Листинг 9. Код программы «Timus\_1496»

**/\***

**Задача #1496**

**Исходные данные:**

**В первой строке записано число N — количество сабмитов в последние 10 минут (0 ≤ N ≤ 100).**

**Следующие N строк содержат названия команд, сабмитивших решения. Названия состоят только из строчных латинских букв и цифр.**

**Длина названий не превосходит 30 символов.**

**Результат:**

**Выведите все аккаунты, под которыми, по мнению жюри, играет спамер. Порядок вывода не важен.**

**\*/**

**import java.io.\*;**

**import java.util.\*;**

**public class Timus\_1496**

**{**

**public static void main(String[] args) throws IOException**

**{**

**BufferedReader f = new BufferedReader(new InputStreamReader(System.in));**

**int n = Integer.parseInt(f.readLine());**

**String[] stores = new String[n];**

**for (int i = 0; i < n; i++)**

**{**

**stores[i] = f.readLine();**

**}**

**Arrays.sort(stores);**

**HashSet<String> duplicates = new HashSet<String>();**

**for (int i = 1; i< n; i++)**

**{**

**if (stores[i].equals(stores[i-1]))**

**{**

**duplicates.add(stores[i]);**

**}**

**}**

**for (String s: duplicates)**

**{**

**System.out.println(s);**

**}**

**f.close();**

**System.exit(0);**

**}**

**}**

Листинг 10. Код программы «Timus\_1581»

**/\***

**Задача #1581**

**Исходные данные:**

**В первой строке находится целое число N — количество чисел, записанных Васей (1 ≤ N ≤ 1000). Во второй строке через пробел записаны эти числа. Все числа целые, положительные и не превосходят 10.**

**Результат:**

**В единственной строке выведите через пробел числа, которые должен был бы записать на бумажке Петя.**

**\*/**

**import java.io.\*;**

**import java.util.\*;**

**public class Timus\_1581**

**{**

**public static void main(String[] args) throws IOException**

**{**

**BufferedReader f = new BufferedReader(new InputStreamReader(System.in));**

**int n = Integer.parseInt(f.readLine());**

**StringTokenizer st = new StringTokenizer(f.readLine());**

**int[] nums = new int[n];**

**for (int i = 0; i < n; i++)**

**{**

**nums[i] = Integer.parseInt(st.nextToken());**

**}**

**int current = 1;**

**int i = 1;**

**int j = 0;**

**while (i < n)**

**{**

**if (nums[i] == nums[j])**

**{**

**current++;**

**}**

**else**

**{**

**System.out.print(current + " " + nums[j] + " ");**

**current = 1;**

**j = i;**

**}**

**i++;**

**}**

**System.out.println(current + " " + nums[j] + " ");**

**f.close();**

**System.exit(0);**

**}**

**}**

Листинг 11. Код программы «Timus\_1639»

**/\***

**Задача #1639**

**Исходные данные:**

**В единственной строке через пробел записаны целые числа m и n (1 ≤ m, n ≤ 50) — длина и ширина**

**шоколадки в дольках.**

**Результат:**

**Если для того, чтобы выиграть, Карлсону нужно ходить первым,**

**выведите в единственной строке «[:=[first]», иначе выведите «[second]=:]».**

**\*/**

**import java.io.\*;**

**import java.util.\*;**

**public class Timus\_1639**

**{**

**static Scanner in = new Scanner(System.in);**

**static PrintWriter out = new PrintWriter(System.out);**

**public static void main(String[] args)**

**{**

**int m = in.nextInt();**

**int n = in.nextInt();**

**int resault = m \* n-1;**

**if ((resault & 1) > 0)**

**{**

**out.println("[:=[first]");**

**}else**

**{**

**out.println("[second]=:]");**

**}**

**out.flush();**

**}**

**}**

Листинг 12. Код программы «Timus\_1753»

**/\***

**Задача # 1753**

**Исходные данные:**

**Единственная строка содержит целые числа h, H и L (1 ≤ h < H < L ≤ 1000).**

**Результат:**

**Выведите с точностью не менее 10−5 величину максимального смещения влево положения центра книги относительно левого края полки.**

**\*/**

**import java.util.Scanner;**

**import static java.lang.Math.\*;**

**public class Timus\_1753**

**{**

**public static void main(String[] args)**

**{**

**Scanner sc = new Scanner(System.in);**

**int h = sc.nextInt();**

**int H = sc.nextInt();**

**int L = sc.nextInt();**

**double x = sqrt(pow(h, 4.0/3.0) \* pow(H, 2.0/3.0)/pow(2, 2.0/3.0) - pow(h, 2.0));**

**double res = H/2.0\*1/sqrt(1 + pow(h/x, 2.0)) - x;**

**if (res <= L && res > 0)**

**{**

**System.out.format("%.6f", res);**

**}**

**else**

**{**

**System.out.format("%.6f", 0.0);**

**}**

**}**

**}**

Листинг 13. Код программы «Timus\_1785»

**/\***

**Задача #1785**

**Исходные данные:**

**В единственной строке записано целое число n (1 ≤ n ≤ 2000) — количество монстров.**

**Результат:**

**Выведите слово, соответствующее данному количеству монстров на языке аниндилъяква.**

**\*/**

**import java.io.PrintWriter;**

**import java.util.Scanner;**

**public class Timus\_1785**

**{**

**public static void main(String[] args)**

**{**

**Scanner in = new Scanner(System.in);**

**PrintWriter out = new PrintWriter(System.out);**

**int a = in.nextInt();**

**if ( a >= 0 && a <= 4)**

**{**

**out.println("few");**

**} if ( a >= 5 && a <= 9)**

**{**

**out.println("several");**

**}if ( a >= 10 && a <= 19)**

**{**

**out.println("pack");**

**}if ( a >= 20 && a <= 49)**

**{**

**out.println("lots");**

**}if ( a >= 50 && a <= 99)**

**{**

**out.println("horde");**

**}if ( a >= 100 && a <= 249)**

**{**

**out.println("throng");**

**}if ( a >= 250 && a <= 499)**

**{**

**out.println("swarm");**

**}if ( a >= 500 && a <= 999)**

**{**

**out.println("zounds");**

**}if ( a >= 1000)**

**{**

**out.println("legion");**

**}**

**out.flush();**

**}**

**}**

Листинг 14. Код программы «Timus\_1787»

**/\***

**Задача #1787**

**Исходные данные:**

**В первой строке записаны целые числа k и n (1 ≤ k, n ≤ 100) — количество машин,**

**успевающих повернуть на «МЕГУ» в течение минуты, и количество минут, прошедших с**

**начала наблюдений. Во второй строке через пробел записаны целые числа a1, …, an (0 ≤ ai ≤ 100),**

**где ai — количество машин, подъехавших к повороту со стороны города в течение**

**i-й минуты. Можно считать, что наблюдения начинаются рано утром, когда машин,**

**ожидающих на повороте, ещё нет.**

**Результат:**

**Выведите количество машин, стоящих в пробке на повороте через n минут**

**после начала наблюдений.**

**\*/**

**import java.util.Scanner;**

**public class Timus\_1787**

**{**

**static Scanner in = new Scanner(System.in);**

**public static void main(String[] args)**

**{**

**int k = in.nextInt();**

**int n = in.nextInt();**

**int car = 0;**

**int [] cars = new int [n];**

**for (int i = 0; i< n; i++)**

**{**

**cars[i] += in.nextInt();**

**}**

**for (int i = 0; i < n; i++)**

**{**

**car += cars[i];**

**if ((car -k)<0 )**

**{**

**car = 0;**

**}else**

**{**

**car -= k;**

**}**

**}**

**System.out.print(car);**

**}**

**}**

Листинг 15. Код программы «Timus\_1820»

**/\***

**Задача #1820**

**Исходные данные:**

**В единственной строке через пробел записаны целые числа n и k (1 ≤ n, k ≤ 1000).**

**Результат:**

**Выведите минимальное количество минут, за которое повар сможет приготовить n бифштексов.**

**\*/**

**import java.io.\*;**

**import java.util.\*;**

**public class Timus\_1820**

**{**

**static Scanner input = new Scanner(System.in);**

**static PrintWriter out = new PrintWriter(System.out);**

**static int calcTime(final int time, int n, int k)**

**{**

**if (n <= 0)**

**{**

**return 0;**

**} else if (n < k)**

**{**

**return 2 \* time;**

**} else**

**{**

**int appender = 0;**

**int n2 = n << 1;**

**if (((n2 % k)) > 0)**

**{**

**appender = 1;**

**}**

**return n2 / k + appender;**

**}**

**}**

**public static void main(String str[])**

**{**

**int n = input.nextInt();**

**int k = input.nextInt();**

**final int time = 1;**

**out.println(calcTime(time, n, k));**

**out.flush();**

**}**

**}**

Листинг 16. Код программы «Timus\_1877»

**/\***

**Задача #1877**

**Исходные данные:**

**В первой строке записан код, установленный на первом замке, во второй строке — код, установленный на втором замке.**

**Оба кода — строки длины 4, состоящие из цифр от 0 до 9.**

**Результат:**

**Выведите «yes», если злоумышленник рано или поздно взломает замок, и «no» в противном случае.**

**\*/**

**import java.io.PrintWriter;**

**import java.util.\*;**

**public class Timus\_1877**

**{**

**static Scanner input = new Scanner(System.in);**

**static PrintWriter out = new PrintWriter(System.out);**

**public static void main(String[] args)**

**{**

**String s1 = input.next();**

**String s2 = input.next();**

**int n1 = Integer.parseInt(s1);**

**int n2 = Integer.parseInt(s2);**

**if ((n1 & 1) == 0 || (n2 & 1) > 0)**

**{**

**out.println("yes");**

**} else**

**{**

**out.println("no");**

**}**

**out.flush();**

**}**

**}**

Листинг 17. Код программы «Timus\_2001»

**/\***

**Задача #2001**

**Исходные данные:**

**Входные данные состоят из трёх строк. В i-й строке (1 ≤ i ≤ 3) записаны целые числа ai и bi**

**(0 ≤ ai, bi ≤ 10 000).**

**Результат:**

**В единственной строке выведите пару целых чисел через пробел — массу ягод, собранных, соответственно,**

**первым и вторым математиком.**

**\*/**

**import java.util.\*;**

**public class Timus\_2001**

**{**

**static Scanner in = new Scanner(System.in);**

**public static void main(String[] args)**

**{**

**int[][] resault = new int[3][2];**

**for (int i = 0; i <= resault.length-1; i++)**

**{**

**for (int j =0; j <= resault[i].length-1; j++)**

**{**

**resault[i][j]= in.nextInt();**

**}**

**}**

**int res = resault[0][0] - resault[2][0];**

**int res\_2 = resault[0][1] - resault[1][1];**

**System.out.println(res);**

**System.out.println(res\_2);**

**}**

**}**

Листинг 18. Код программы «Timus\_2012»

**/\***

**Задача #2012**

**Исходные данные:**

**В единственной строке записано целое число f — сколько задач хочет решить Гриша за первый час соревнования (1 ≤ f ≤ 11).**

**Результат:**

**Выведите «YES», если Грише в одиночку удастся решить все предложенные задачи, и «NO» в противном случае.**

**\*/**

**import java.io.PrintWriter;**

**import java.util.Scanner;**

**public class Timus\_2012**

**{**

**public static void main(String[] args)**

**{**

**Scanner in = new Scanner(System.in);**

**PrintWriter out = new PrintWriter(System.out);**

**int a = in.nextInt();**

**int time = 45;**

**int totalTime = 240;**

**if (a>= 1 && a <= 11)**

**{**

**int oneTime = 12 - a;**

**int remainingTime = time \* oneTime;**

**if (remainingTime <= totalTime)**

**{**

**out.println("YES");**

**}else**

**{**

**out.println("NO");**

**}**

**}**

**out.flush();**

**}**

**}**

Листинг 19. Код программы «Timus\_2066»

**/\***

**Задача #2066**

**Исходные данные:**

**Даны целые неотрицательные числа a, b, c в неубывающем порядке**

**(0 ≤ a ≤ b ≤ c ≤ 100), каждое в отдельной строке.**

**Результат:**

**Выведите одно целое число — минимальное значение выражения.**

**\*/**

**import java.util.\*;**

**public class Timus\_2066**

**{**

**static Scanner in = new Scanner(System.in);**

**public static void main(String[] args)**

**{**

**int a = in.nextInt();**

**int b = in.nextInt();**

**int c = in.nextInt();**

**if (b==0 || b == 1 || c ==1 )**

**{**

**System.out.print(a-b-c);**

**}else**

**{**

**System.out.print(a-b\*c);**

**}**

**}**

**}**

Листинг 20. Код программы «Timus\_1224»

**/\***

**Задача #1224**

**Исходные данные:**

**Ввод состоит из двух целых чисел в следующем порядке: N, M (1 ≤ N, M ≤ 231 − 1).**

**Результат:**

**Вывод состоит из одного целого числа — количества поворотов.**

**\*/**

**import java.util.Scanner;**

**public class Timus\_1224**

**{**

**public static long turns(long N, long M)**

**{**

**if (N <= M)**

**{**

**return 4\*(N/2) - 2\*((N + 1) % 2);**

**}**

**return 4\*(M/2) - 2\*((M + 1) % 2) + 1;**

**}**

**public static void main(String[] args)**

**{**

**Scanner sc = new Scanner(System.in);**

**long N = sc.nextLong();**

**long M = sc.nextLong();**

**System.out.println(turns(N, M));**

**}**

**}**

5. Вывод: в процессе написания 20 программ по задачам с сайта <https://acm.timus.ru/> укрепил навыки работы на языке программирования Java, а также усвоил новые методы, библиотеки и повторил работу с классами.

GitHub - <https://github.com/Ve4noMolodoi/JavaLabExample/tree/main/Timus/src>.