# 7-8 класс

# Задача 1. Минимальный делитель

*Имя входного файла:* **input.txt**

*Имя выходного файла:* **output.txt**

*Максимальное время работы на одном тесте: 1 секунда*

Дано целое число, не меньшее 2. Выведите его наименьший натуральный делитель, отличный от 1.

## Формат входного файла

Первая строка входного файла содержит число *N*– целое число.

## Формат выходного файла

Выведите число, которое является ответом на задачу.

## Пример

|  |  |
| --- | --- |
| **input.txt** | **output.txt** |
| 15 | 3 |
| 2 | 2 |
| 3 | 3 |

# Задача 2. Коробки

*Имя входного файла: input.txt*

*Имя выходного файла: output.txt*

*Максимальное время работы на одном тесте: 1 секунда*

Есть две коробки, первая размером A₁×B₁×C₁, вторая размером A₂×B₂×C₂. Определите, можно ли разместить одну из этих коробок внутри другой, при условии, что поворачивать коробки можно только на 90 градусов вокруг ребер.

## Формат входного файла

Во входном файле числа A₁,B₁,C₁,A₂,B₂,C₂.

## Формат выходного файла

В выходной файл программа должна вывести одну из следующих строчек:

Boxesareequal, если коробки одинаковые, Thefirstboxissmallerthanthesecondone, если первая коробка может быть положена во вторую, Thefirstboxislargerthanthesecondone, если вторая коробка может быть положена в первую, Boxesareincomparable, во всех остальных случаях.

## Примеры

|  |  |
| --- | --- |
| **input.txt** | **output.txt** |
| 123321 | Boxesareequal |
| 223321 | The first box is larger than the second one |

# Задача 3. Школьное время

*Имя входного файла:* **input.txt**

*Имя выходного файла:* **output.txt**

*Максимальное время работы на одном тесте: 1 секунда*

Вот так выглядит расписание звонков, которое вы можете найти в школе:

1. 08:30 9:10

2. 9:20 10:00

3. 10:30 11:10

4. 11:20 12:00

5. 12:20 13:00

6. 13:10 13:50

7. 14:00 14:40

Дорога от дома до школы, как и дорога назад, занимает у Кирилла ровно 10 минут. После прихода из дома на первый урок в своём расписании Кирилл сидит в школе до конца последнего урока в своём расписании, а потом идет назад домой. Посчитайте, сколько времени Кирилл суммарно потратит на то, чтобы сходить в школу, посетить там все уроки и вернуться домой.

## Формат входного файла

Во входном файле записаны два целых числа от 1 до 7 номера первого и последнего урока Кирилла. Номер первого урока не превосходит номера последнего урока.

## Формат выходного файла

Выведите суммарное значение времени, затраченное Кириллом на дорогу и посещение школы.

## Пример

|  |  |
| --- | --- |
| **input.txt** | **output.txt** |
| 13 | 7:00 |
| 37 | 9:50 |

# Задача 4. Олимпиада

*Имя входного файла:* **input.txt**

*Имя выходного файла:* **output.txt**

*Максимальное время работы на одном тесте: 1 секунда*

В олимпиаде участвовало N человек. Каждый получил определенное количество баллов, при этом оказалось, что у всех участников разное число баллов. Упорядочите список участников олимпиады в порядке убывания набранных баллов.

## Формат входного файла

Программа получает на вход число участников олимпиады N. Далее идет N строк, в каждой строке записана фамилия участника, затем, через пробел, набранное им количество баллов.

## Формат выходного файла

Выведите список участников (только фамилии) в порядке убывания набранных баллов.

## Пример

|  |  |
| --- | --- |
| **input.txt** | **output.txt** |
| 3Ivanov 15Petrov 10Sidorov 20 | SidorovIvanovPetrov |
| 3Ivanov 15Petrov 20Sidorov 10 | PetrovIvanovSidorov |
| 3Ivanov 10Petrov 15Sidorov 20 | SidorovPetrovIvanov |

# 9-11 класс

# Задача 1. Возведение в степень

*Имя входного файла:* **input.txt**

*Имя выходного файла:* **output.txt**

*Максимальное время работы на одном тесте: 1 секунда*

Возводить в степень можно гораздо быстрее, чем за n умножений! Для этого нужно воспользоваться следующими рекуррентными соотношениями: aⁿ = (a²)ⁿ/² при четном n, aⁿ=a⋅aⁿ⁻¹ при нечетном n. Реализуйте алгоритм быстрого возведения в степень. Если вы все сделаете правильно,то сложность вашего алгоритма будет O(logn).

## Формат входного файла

В первой строке входного файла задано действительное число a и целое неотрицательное число n.

## Формат выходного файла

Результат возведения числа в степень.

## Примеры

|  |  |
| --- | --- |
| **input.txt** | **output.txt** |
| 21 | 2 |
| 22 | 4 |
| 23 | 8 |

# Задача 2. Арифметический палиндром

*Имя входного файла: input.txt*

*Имя выходного файла: output.txt*

*Максимальное время работы на одном тесте: 1 секунда*

Арифметический палиндром представляет собой положительное число, которое одинаково считывается вперед и назад. Например, 87578 и 123321 это арифметические палиндромы, а 3753 и 81128 нет. Требуется найти количество способов представить число n как сумму двух палиндромов.

## Формат входного файла

В первой строке находится натуральное число n (1 ≤ n ≤ 105).

## Формат выходного файла

Выведите результат способы представления на разных строчках, в последней строке выведите количество способов

## Примеры

|  |  |
| --- | --- |
| **input.txt** | **output.txt** |
| 55 | 332211442 способа |
| 242  | 141101131 1112 способа |
| 121  | 99 2288 3377 4466 554 способа |

# Задача 3. Океан

*Имя входного файла:* **input.txt**

*Имя выходного файла:* **output.txt**

*Максимальное время работы на одном тесте: 1 секунда*

Океан можно представить в виде I четверти декартовой плоскости. В океане каждая рыба имеет координаты (a,b) и в одной точке на плоскости может быть несколько рыб. Также есть рыбаки, у каждого рыбака есть значение только x координаты (значение координаты y для всех рыбаков равно 0), у каждого рыбака есть удочка заданной длиной L (для всех рыбаков одинаковое значение). Рыбак может поймать рыбу на расстоянии меньшем либо равном L. Расстояние между рыбаком и рыбой можно найти по формуле r=|a-x|+b. Найти сколько рыбы могут поймать все рыбаки, если рыба доступна сразу нескольким рыбакам, то она считается пойманной для всех.

## Формат входного файла

Первая строка ввода содержит три целых числа n, m, и число рыб, количество рыбаков и длина удочки соответственно. Каждая из следующих n строк содержит два целых числа x и y координаты рыб. Следующая строка содержит m целых чисел, координаты рыбаков

## Формат выходного файла

Выходной файл содержит одно число – искомое количество.

## Пример

|  |  |
| --- | --- |
| **input.txt** | **output.txt** |
| 5 3 41 12 23 45 74 03 4 7 | 8 |

# Задача 4. Проходной балл

*Имя входного файла: input.txt*

*Имя выходного файла: output.txt*

*Максимальное время работы на одном тесте: 1 секунда*

Для поступления в вуз абитуриент должен предъявить результаты трех экзаменов в виде ЕГЭ, каждый из них оценивается целым числом от 0 до 100 баллов. При этом абитуриенты, набравшие менее 40 баллов (неудовлетворительную оценку) по любому экзамену из конкурса выбывают. Остальные абитуриенты участвуют в конкурсе по сумме баллов за три экзамена.

В конкурсе участвует N человек, при этом количество мест равно K. Определите проходной балл, то есть такое количество баллов, что количество участников, набравших столько или больше баллов не превосходит K, а при добавлении к ним абитуриентов, набравших наибольшее количество баллов среди непринятых абитуриентов, общее число принятых абитуриентов станет больше K.

## Формат входного файла

## Программа получает на вход количество мест K. Далее идут строки с информацией об абитуриентах, каждая из которых состоит из имени (текстовая строка, содержащая произвольное число пробелов) и трех чисел от 0 до 100, разделенных пробелами.Используйте для ввода файл input.txt с указанием кодировки utf8.

## Формат выходного файла

## Программа должна вывести проходной балл в конкурсе. Выведенное значение должно быть минимальным баллом, который набрал абитуриент, прошедший по конкурсу.

## Также возможны две ситуации, когда проходной балл не определен.

## Если будут зачислены все абитуриенты, не имеющие неудовлетворительных оценок, программа должна вывести число 0.

## Если количество имеющих равный максимальный балл абитуриентов больше, чем K, программа должна вывести число 1.

## Используйте для вывода файл output.txt с указанием кодировки utf8.

## Примеры

|  |  |
| --- | --- |
| **input.txt** | **output.txt** |
| 5Иванов Сергей 70 70 70Сергеев Петр 100 100 0Петров Василий 70 60 70Васильев Андрей 70 60 70Андреев Денис 100 30 100Денисов Роман 50 50 50Романов Иван 60 70 70Ким Чен Ир 50 50 50Ким Ир Сен 40 40 40 | 200 |
| 1Иванов Сергей 40 40 40Сергеев Петр 100 100 39 | 0 |
| 1Иванов Сергей 60 60 60Сергеев Петр 100 40 40 | 1 |

# Задача 1Минимальныйделитель(100б.)

import math

defMinDivisor(n):

i = 2

 d = 0

 while (i<= int(n\*\*(1/2))):

 if n % i == 0:

 d = i

 break

i = i + 1

 if d == 0:

 d = n

 return d

n = int(input())

print(MinDivisor(n))

# Задача 2Коробки(100б.)

a, b, c = int(input()), int(input()), int(input())

x, y, z = int(input()), int(input()), int(input())

n = 0

if ((x == a) and (y == b) and (z == c)):

 n = n + 1

if ((x == b) and (y == c) and (z == a)):

 n = n + 1

if ((x == c) and (y == b) and (z == a)):

 n = n + 1

if n > 0:

print('Boxes are equal')

else:

 if a > b:

 (a, b) = (b, a)

 if b > c:

 (b, c) = (c, b)

 if a > b:

 (a, b) = (b, a)

 if x > y:

 (x, y) = (y, x)

 if y > z:

 (y, z) = (z, y)

 if x > y:

 (x, y) = (y, x)

 if ((x <= a) and (y <= b) and (z <= c)):

print('The first box is larger than the second one')

 else:

 n = n + 1

 if ((x >= a) and (y >= b) and (z >= c)):

print('The first box is smaller than the second one')

 else:

 n = n + 1

 if n > 1:

print('Boxes are incomparable')

# Задача 3Школьноевремя(100б.)

classProgram

 {

staticvoid Main(string[] args)

 {

varstr = Console.ReadLine().Split(' ');

int p1 = Convert.ToInt32(str[0]);

int p2 = Convert.ToInt32(str[1]);

intotvet = 20 + (40 \* (p2 - p1 + 1) + 10 \* (p2 - p1));

if (p1 <= 2 && p2 >= 3) otvet += 20;

if (p1 <= 4 && p2 >= 5) otvet += 10;

Console.WriteLine(otvet / 60 + ":" + otvet % 60);

Console.ReadKey();

}

 }

# Задача 4Олимпиада(100б.)

num = int(input())

members = [input().split() for \_ in range(num)]

for l in sorted(members, key=lambda x: int(x[1]), reverse=True):

print(l[0])

# Задача 1 Возведение в степень(100б.)

importmath

def power(a, n):

 if n == 0:

 return 1

 if n == 1:

 return a

 return a \* power(a, n - 1)

a = float(input())

n = int(input())

print(power(a, n))

# Задача 2 Арифметический палиндром(100б.)

classProgram

 {

privatestaticintreverse(int num)

 {

int acc = 0;

while (num > 0)

 {

 acc = acc \* 10 + num % 10;

 num /= 10;

 }

return acc;

 }

staticvoid Main(string[] args)

 {

varKol = 0;

var N = Convert.ToInt32(Console.ReadLine());

for (inti = 1; i< N/2; i++)

 {

if (i == reverse(i))

 {

var j = N - i;

if (reverse(j) == j)

 {

Kol++;

Console.WriteLine(j + " " + i);

 }

 }

 }

Console.WriteLine(Kol);

Console.ReadKey();

 }

 }

# }

# Задача 3Океан(100б.)

class Program

 {

 static void Main(string[] args)

{

Console.WriteLine("Введите N-число рыб, K-количество рыбаков,L-длина удочки");

vararr = (Console.ReadLine().Split(' '));

 var N = Convert.ToInt32(arr[0]);

 var K = Convert.ToInt32(arr[1]);

 var L = Convert.ToInt32(arr[2]);

int[] fishX = new int[N];

int[] fishY = new int[N];

Console.WriteLine("Введитекоординаты " + N + " рыб");

 for (inti = 0; i< N; i++)

 {

vartmp = Console.ReadLine().Split(' ');

fishX[i] = Convert.ToInt32(tmp[0]);

fishY[i] = Convert.ToInt32(tmp[1]);

 }

int[] fisherman = new int[K];

Console.WriteLine("Введитекоординату Y для " + K + " рыбаков");

arr = Console.ReadLine().Split(' ');

 for (inti = 0; i< K; i++)

 {

 fisherman[i] = Convert.ToInt32(arr[i]);

 }

varKol = 0;

 for (inti = 0; i< K; i++)

 {

 for (int j = 0; j < N; j++)

 {

 if (Math.Abs(fishX[j] - fisherman[i]) + fishY[j] <= L)

 {

Kol++;

 }

 }

 }

Console.WriteLine("Количествопойманныхрыб=" + Kol);

Console.ReadKey();

 }

 }

# Задача 4Проходнойбалл(100б.)

myFile = open("input.txt", "r", encoding="utf8")

k = int(myFile.readline())

myList = []

for line in myFile:

newLine = line.split()

 if int(newLine[-1]) >= 40 and int(newLine[-2]) >= 40 \

 and int(newLine[-3]) >= 40:

myList.append(newLine)

myFile.close()

myList.sort(key=lambda a: int(a[-1]) + int(a[-2]) + int(a[-3]))

myList.reverse()

konk = []

for i in myList:

 sum = int(i[-1]) + int(i[-2]) + int(i[-3])

konk.append(sum)

n = len(konk)

defkonkurs(n, k, konk):

 if n <= k:

 return 0

elifkonk[k] == konk[0]:

 return 1

 for i in range(k, 0, -1):

 if konk[i] <konk[i - 1]:

 return konk[i - 1]

print(konkurs(n, k, konk))