

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Профиль Информационные технологии**  
**Междисциплинарные задачи**

---

**I Вариант**

**Задача 1 (20 баллов)**

В процессе изучения систем счисления Петя заметил, что в каждой из них есть определенные наборы цифр (и не только), при помощи которых записывается то или иное число. Особенно его заинтересовала сумма цифр в представлении обычного числа.

Помогите Пете найти сумму цифр числа в системе счисления с основанием 4. Например, число 23 в этой системе будет представлено как 113, и верным ответом будет 5.

Ограничения

- Для решения необходимо использовать стандартные потоки ввода вывода.
- На вход обязательно будет подана ровно одна строка с единственным числом.
- Подаваемое число находится в промежутке  $[0, 1000000]$

Примеры ввода/вывода

<b>Ввод</b>	<b>Вывод</b>
0	0
23	5
43987	16

**Пример решения (C#/.NET 5)**

```
using System;

namespace Task1.Var1
{
    class Program
    {
        static void Main(string[] args)
        {
            var value = int.Parse(Console.ReadLine());
            var sum = 0;
            while (value > 0)
            {
                sum += value % 4;
                value /= 4;
            }
            Console.WriteLine(sum);
        }
    }
}
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Профиль Информационные технологии**  
**Междисциплинарные задачи**

---

```
}  
}  
}
```

**Критерии оценивания:**

- 0 - решение не прошло тестирование;
- 6 - решение прошло только тесты базовой группы;
- 12 - решение прошло тесты базовой и средней групп;
- 20 - решение прошло все тесты.

**Задача 2 (30 баллов)**

Пети нашел калькулятор, но оказалось, что тот сломан. Методом проб и ошибок выяснилось, что калькулятор умеет делать только три возможных действия – прибавить 12, вычесть 8, умножить на 2. Изначально на калькуляторе выставлено число “1”, и изменить этого никак нельзя. При этом при попытке получить отрицательное число при вычитании калькулятор выключается.

Необходимо узнать, возможно ли на этом калькуляторе вывести определенное число за максимум 100 операций. Например, число 13 вывести можно – выполнить действие “прибавить 12”. Как и число “96” – “прибавить 12”, “умножить на 2”, “умножить на 2”, “умножить на 2”, “вычесть 8”.

Искомое число вводится в единственной строке. На вывод необходимо вывести “NO” если число получить представленными операциями невозможно, и “YES” если возможно.

**Ограничения**

- Для решения необходимо использовать стандартные потоки ввода вывода.
- На вход обязательно будет подана ровно одна строка с единственным числом.
- Подаваемое число находится в промежутке [0, 1000000]

**Примеры ввода/вывода**

<b>Ввод</b>	<b>Вывод</b>
0	YES

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Профиль Информационные технологии**  
**Междисциплинарные задачи**

9	YES
11	NO

**Критерии оценивания:**

- 0 - решение не прошло тестирование;
- 10 - решение прошло только тесты базовой группы;
- 20 - решение прошло тесты базовой и средней группы;
- 30 - решение прошло все тесты;

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Профиль Информационные технологии**  
**Междисциплинарные задачи**

---

**Пример решения (C#/.NET 5)**

```
using System;
using System.Collections.Generic;

namespace Task2.Var2
{
    class Program
    {
        private static readonly HashSet<int> broken = new HashSet<int>();
        static void Main(string[] args)
        {
            int value = int.Parse(Console.ReadLine());
            Console.WriteLine(IsPossible(1, value, 0) ? "YES" : "NO");
        }
        private static bool IsPossible(int value, int targetValue, int deep)
        {
            if (value == targetValue) return true;
            if (value < 0) return false;
            if (deep == 100) return false;
            if (broken.Contains(value)) return false;

            var result = IsPossible(value + 12, targetValue, deep + 1) ||
                IsPossible(value - 8, targetValue, deep + 1) ||
                IsPossible(value * 2, targetValue, deep + 1);
            if (result == true)
            {
                return true;
            }
            else
            {
                broken.Add(value);
                return result;
            }
        }
    }
}
```

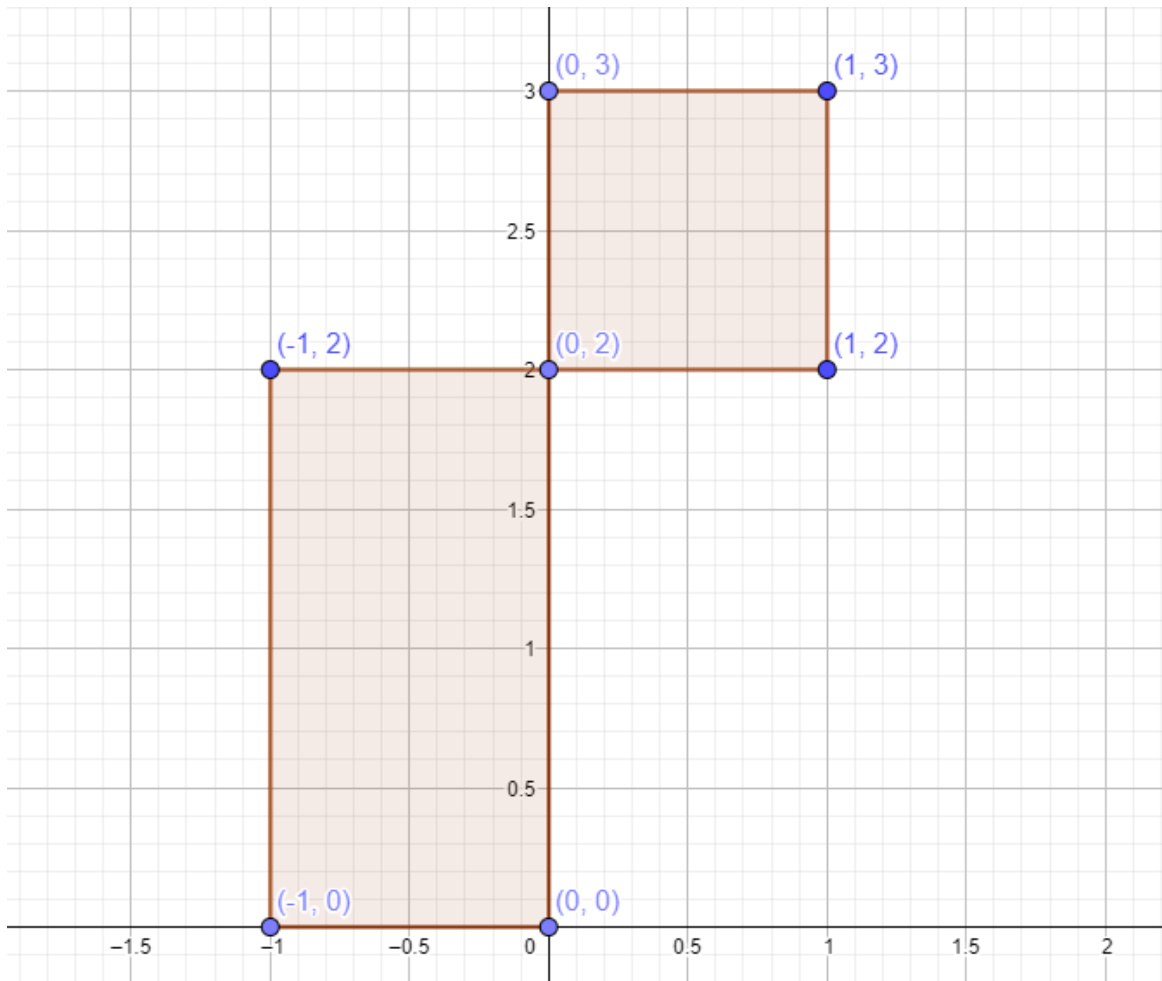
**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
Заключительный этап  
Профиль Информационные технологии  
Междисциплинарные задачи

---

**Задача 3 (50 баллов)**

Задаются координаты двух прямоугольников на плоскости. Необходимо выяснить, пересекаются ли они. Пересечением называется наложение площадей, сторон или вершин прямоугольников. Если прямоугольники пересекаются – необходимо найти площадь пересечения.

Следующие прямоугольники являются пересекающимися:

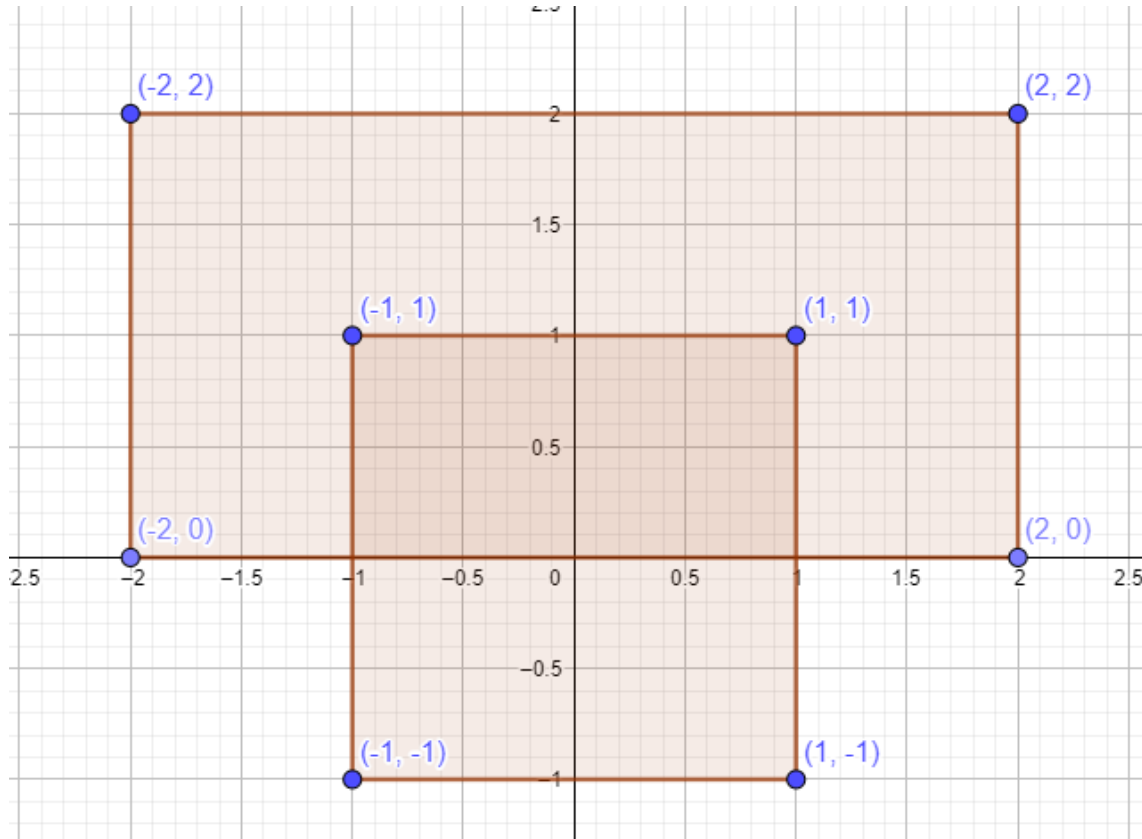


Для представленных выше фигур площадь пересечения будет равна 0.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
Заключительный этап  
**Профиль Информационные технологии**  
**Междисциплинарные задачи**

---

Для следующих прямоугольников площадь пересечения будет равна 2:



Каждый прямоугольник задается как последовательность из четырех координат – пар целых чисел  $X_i Y_i$ , разделенных пробелами. Перечисление координат начинается с нижней правой координаты, и происходит против часовой стрелки. Координаты каждого прямоугольника записаны в отдельной строке.

Например, координаты прямоугольников на предыдущем изображении будут представлены следующим образом:

```
2 0 2 2 -2 2 -2 0
```

```
1 -1 1 1 -1 1 -1 -1
```

На выход необходимо подать единственное число – площадь пересечения прямоугольников. Если прямоугольники не пересекаются – необходимо вывести -1.

Ограничения

- Для решения необходимо использовать стандартные потоки ввода вывода.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Профиль Информационные технологии**  
**Междисциплинарные задачи**

---

- На вход обязательно будут поданы ровно две строки, в каждой по 8 чисел.
- Подаваемые координаты обязательно формируют прямоугольник.
- Все координаты представлены целыми числами в промежутке [-10000, 10000]

Примеры ввода/вывода

Ввод	Вывод
2 0 2 2 -2 2 -2 0 1 -1 1 1 -1 1 -1 -1	2
0 0 0 6 -5 6 -5 0 -2 1 -2 5 -4 5 -4 1	8
0 0 0 9 -1 9 -1 0 2 5 2 6 1 6 1 5	-1

**Пример решения (C#/.NET 5)**

```
using System;
using System.Linq;

namespace Task3.Var2
{
    class Program
    {
        static void Main(string[] args)
        {
            var rect1 = Console.ReadLine().Split(" ").Select(int.Parse).ToArray();
            var rect2 = Console.ReadLine().Split(" ").Select(int.Parse).ToArray();
            Console.WriteLine(Intersection(rect1, rect2));
        }

        /*
         * rectangle points in array
         * [4][5] ..... [2][3]
         * .....
         * .....
         * [6][7] ..... [0][1]
         */
        static int Intersection(int[] rect1, int[] rect2)
        {
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**

**Заключительный этап**

**Профиль Информационные технологии  
Междисциплинарные задачи**

---

```
int x1 = Math.Max(rect1[4], rect2[4]);
int x2 = Math.Min(rect1[2], rect2[2]);
int y1 = Math.Min(rect1[5], rect2[5]);
int y2 = Math.Max(rect1[7], rect2[7]);

if (x2 >= x1 && y2 <= y1)
{
    return (x2 - x1) * (y1 - y2);
}
return -1;
}
}
```

**Критерии оценивания:**

- 0 - решение не прошло тестирование;
- 16 - решение прошло только тесты базовой группы;
- 32 - решение прошло тесты базовой и средней групп;
- 50 - решение прошло все тесты;



**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Профиль Информационные технологии**  
**Междисциплинарные задачи**

---

**II Вариант**

**Задача 1 (20 баллов)**

На входе сначала строка с числом  $N$  ( $2 < N < 6$ ), а потом строка с числом  $K$  ( $0.5 < K < 0.91$ ). Данные числа представляют собой  $N$  - количество пакетов, которые некоторая станция передает серверу,  $K$  - вероятность успешной передачи каждого пакета данных.

В сети располагают регистрирующее устройство, которое может генерировать два сообщения:  $x_1$  – все пакеты переданы успешно,  $x_2$  – ошибка при передаче хотя бы одного пакета данных.

Входные данные не гарантируются, программа должна завершаться при некорректном вводе и вывести «Repeat».

Необходимо найти:

1. Вероятность сообщения  $X_2$
2. Среднее количество информации в сообщении о результате передачи пакетов данных

Формат выходных данных соблюдать как показано в примере, количество знаков после запятой не должно превышать 4.

Пример входных данных	Пример выходных данных
5	0.1846
0.96	0.6901

**Пример решения:**

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <cmath>
#include <algorithm>

using namespace std;
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Профиль Информационные технологии**  
**Междисциплинарные задачи**

---

```
int main() {
int n = 0;
    if (!(std::cin >> n)) return 1;
    float usp = 0;
    if (!(std::cin >> usp)) return 1;

// вероятность что все будет успешно
// usp^n
float x1;
x1 = powf(usp, n);

    // неуспешное
float x2;
    x2 = 1 - x1;
std::cout << x2;
std::cout << "\n";
//энтропия
float entrop;
entrop = x1*log2(1/x1) + x2*log2(1/x2);
printf("%.4f", entrop);
return 0;
}
```

**Критерии оценивания:**

- 0 - решение не прошло тестирование;
- 6 - решение прошло только тесты базовой группы;
- 12 - решение прошло тесты базовой и средней групп;
- 20 - решение прошло все тесты.

**Задача 2 (30 баллов)**

Дима и Лиля любят играть с цифрами. В этот раз у них получилась такая зависимость:  
 $a \cdot 9 + b = c$ , где  $c$  может принимать значения: 11, 111, 1111 и так далее.

Напишите программу, которой на вход подадут значение  $c$ , на выход выдается значение  $a$  и  $b$  через пробел.

Примеры прогонов:

Вход	Выход
------	-------

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Профиль Информационные технологии**  
**Междисциплинарные задачи**

---

11	1 2
111	12 3

**Пример решения:**

Если посмотреть на входные значения, то можно увидеть определённую закономерность.

$$1 * 9 + 2 = 11$$

$$12 * 9 + 3 = 111$$

$$123 * 9 + 4 = 1111$$

$$1234 * 9 + 5 = 11111$$

$$12345 * 9 + 6 = 111111$$

$$123456 * 9 + 7 = 1111111$$

$$1234567 * 9 + 8 = 11111111$$

$$12345678 * 9 + 9 = 111111111$$

$$123456789 * 9 + 10 = 1111111111$$

Данную пирамиду можно найти под названием «Ряд единиц через цифру 9». По условию задачи на вход поступает результат последовательности, а программа должна выдать на выходе множимое и второе слагаемое.

Представив пирамиду можно увидеть, что вторые слагаемые являются последовательностью от 2 до 10. Так же второе слагаемое отражает количество цифр в результате последовательности. Способ расчета остается за участником олимпиады, но оба они являются верными.

В вопросе же с множимым все немного сложнее. Множимое представляет собой последовательность чисел длиной на единицу меньшей, чем результат последовательности, основанной на последовательности цифр «123456789».

Соответственно в примерах ввода и вывода можно в этом убедиться: «11» и «1»; «111» и «12».

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Профиль Информационные технологии**  
**Междисциплинарные задачи**

---

**Пример правильного решения на языке программирования Python 3.7.4**

```
c = int(input())
length = str(c).count("1")
a = ""
b = 1
for i in range(1, length):
    a += str(i)
    b += 1
print(a, b)
```

**Критерии оценивания:**

- 0 - решение не прошло тестирование;
- 10 - решение прошло только тесты базовой группы;
- 20 - решение прошло тесты базовой и средней групп;
- 30 - решение прошло все тесты;

**Задача 3 (50 баллов)**

- Имя входного файла:                      стандартный ввод
- Имя выходного файла:                      стандартный вывод
- Ограничение по времени:                  1 секунда
- Ограничение по памяти: 256 мегабайт

Ваня - непутевый студент. Учась на удаленке, сильно расслабился и не смог сдать экзамен по теории игр. Но его преподаватель решил дать Ване шанс получить тройку. Он предложил игру: задается какая-то дата 2021 года. Каждый игрок задает на своем ходе какую-нибудь более позднюю дату, увеличивая на 1 или 2 либо день, либо месяц, но только не и то и другое сразу. При этом сочетание дня и месяца должно оставаться датой в 2021. Игрок, назвавший 31 декабря, проиграл. Ваня, как и преподаватель, играет наилучшим образом. Если Ваня побеждает - получает заветную тройку. Исходя из заданной начальной

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Заключительный этап**  
**Профиль Информационные технологии**  
**Междисциплинарные задачи**

---

даты, помогите понять, сможет ли Ваня выиграть и не завалить экзамен, если он ходит первым.

Формат входных данных

Два целых числа -  $d$  и  $m$  - день и месяц в 2021 ( $1 \leq d \leq 31$ ,  $1 \leq m \leq 12$ ).

Формат выходных данных

Ответ на вопрос задачи - «YES», если Ваня сможет выиграть, «NO» в ином случае.

Примеры

стандартный ввод	стандартный вывод
28 12	«YES»
27 12	«NO»

Замечание

Если задана дата 28.12, то Ваня может назвать 30.12 и преподавателю останется назвать 31.12 и значит, что Ваня выигрывает.

**Пример решения:**

```
def is_valid_date(day, month):
    if day < 1 or day > 31 or month < 1 or month > 12:
        return False
    if month == 2:
        return day <= 28
    if month in [1, 3, 5, 7, 8, 10, 12]:
        return day <= 31
    return day <= 30

def main():
    d0, m0 = map(int, input().split())

    if d0 == 31 and m0 == 12:
        print("NO")
        return

    win = [[False] * 20 for i in range(40)]
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**

**Заключительный этап**

**Профиль Информационные технологии  
Междисциплинарные задачи**

---

```
win[31][12] = True
m = 12
while m >= 1:
    d = 31
    while d >= 1:
        if d == 31 and m == 12:
            d = d - 1
            continue
        is_winning = False
        if is_valid_date(d + 1, m) and win[d + 1][m] is False:
            is_winning = True
        if is_valid_date(d + 2, m) and win[d + 2][m] is False:
            is_winning = True
        if is_valid_date(d, m + 1) and win[d][m + 1] is False:
            is_winning = True
        if is_valid_date(d, m + 2) and win[d][m + 2] is False:
            is_winning = True
        win[d][m] = is_winning
        d = d - 1
    m = m - 1

if win[d0][m0] is True:
    print("YES")
else:
    print("NO")
```

main()

**Критерии оценивания:**

- 0 - решение не прошло тестирование;
- 16 - решение прошло только тесты базовой группы;
- 32 - решение прошло тесты базовой и средней групп;
- 50 - решение прошло все тесты;