1. (**П. Волгин**) Алгоритм вычисления функции *F*(*n*), где *n* – целое неотрицательное число, задан следующими соотношениями:

*F*(*n*) = 1 при *n* ≤ 1

*F*(*n*) = 2·*F*(*n* – 1) + *F*(*n –* 2), если *n* > 1 и *n* кратно 3,

*F*(*n*) = 3·*F*(*n* – 2) + *F*(*n –* 1) в остальных случаях.

Сколько существует значений *n* на отрезке [1, 35], для которых сумма цифр значения функции *F*(*n*) является простым числом?

1. (**А. Куканова**) Алгоритм вычисления функции *F*(*n*), где *n* – натуральное число, задан следующими соотношениями:

*F*(*n*) = 1, если *n* = 1

*F*(*n*) = *n* · *F*(*n* – 1) *+* 1, если *n* > 1.

Чему равно значение выражения *F*(3303) / *F*(3300)? В ответе укажите только целую часть числа.

1. (**PRO100 ЕГЭ**) Обозначим частное от деления натурального числа a на натуральное число b как a // b, а остаток как a%b. Алгоритм вычисления функции *F*(*n*), где *n* – натуральное число, задан следующими соотношениями:

*F*(*n*) *= n,* при *n ­* <10,

*F*(*n*) *=* (*n* % 10)· *F*(*n//*10)*,* если *n* ≥10*.*

Найдите количество чисел *n* из отрезка [1 000 000 000 000; 9 999 999 999 999], для которых F(n) не равно нулю.

1. (**А. Богданов**) Обозначим операцию целочисленного деления с округлением вниз как «//», а нахождения остатка деления через «%». Например, 8 // 3 == 2 и 7 % 3 == 1. Алгоритм вычисления функции F(n), где n – натуральное число, задан следующими соотношениями:

*F*(*n*) *= n,* если *n* <2*,*

*F*(*n*) *= F*(*n­ //* 2) + *F*(*n­* %2)*,* если *n ≥* 2*.*

1. Определите, сколько символов \* выведет эта процедура при вызове F(28):

|  |  |  |
| --- | --- | --- |
| **Python** | **Паскаль** | **C++** |
| **def F( n ):**  **print('\*')**  **if n >= 1:**  **print('\*')**  **F(n-1)**  **F(n-2)** | **procedure F( n: integer );**  **begin**  **write('\*');**  **if n >= 1 then begin**  **write('\*');**  **F(n-1);**  **F(n-2);**  **end;**  **end;** | **void F( int n )**  **{**  **cout << '\*';**  **if( n >= 1 ) {**  **cout << '\*';**  **F(n-1);**  **F(n-2);**  **}**  **}** |

1. Определите наименьшее значение *n*, при котором сумма чисел, которые будут выведены при вызове *F*(*n*), будет больше 1000000. Запишите в ответе сначала найденное значение *n*, а затем через пробел – соответствующую сумму выведенных чисел*.*

|  |  |  |
| --- | --- | --- |
| **Python** | **Паскаль** | **C++** |
| **def F( n ):**  **print(n+1)**  **if n > 1:**  **print(n+5)**  **F(n-1)**  **F(n-2)** | **procedure F**  **( n: integer );**  **begin**  **writeln(n+1);**  **if n > 1 then begin**  **writeln(n+5);**  **F(n-1);**  **F(n-2);**  **end;**  **end;** | **void F( int n )**  **{**  **cout << n+1 << endl;**  **if( n > 1 ) {**  **cout << n+5 << endl;**  **F(n-1);**  **F(n-2);**  **}**  **}** |

**Р-07 (демо-2021).** Текстовый файл **24.txt** состоит не более чем из 106 символов X, Y и Z. Определите максимальное количество идущих подряд символов, среди которых каждые два соседних различны. Для выполнения этого задания следует написать программу.

**Р-05.** В текстовом файле **k8.txt** находится цепочка из символов, в которую могут входить заглавные буквы латинского алфавита A…Z и десятичные цифры. Найдите длину самой длинной подцепочки, состоящей из одинаковых символов. Выведите сначала символ, из которого строится цепочка, а затем через пробел – длину этой цепочки.

**Р-02. (А.М. Кабанов)** В текстовом файле **k7.txt** находится цепочка из символов латинского алфавита A, B, C, D, E. Найдите длину самой длинной подцепочки, состоящей из символов A, B или C (в произвольном порядке).

**Р-04 (Демо-вариант 2023).** *Назовём маской числа последовательность цифр, в которой также могут встречаться следующие символы:*

*– символ «?» означает ровно одну произвольную цифру;*

*– символ «\*» означает любую последовательность цифр произвольной длины; в том числе «\*» может задавать и пустую последовательность. Например, маске 123\*4?5 соответствуют числа 123405 и 12300405.*

*Среди натуральных чисел, не превышающих 1010, найдите все числа, соответствующие маске 1?2139\*4, делящиеся на 2023 без остатка.*

*В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 2023.*

1. При проведении эксперимента заряженные частицы попадают на чувствительный экран, представляющий из себя матрицу размером 10000 на 10000 точек. При попадании очередной частицы на экран в файл записываются координаты чувствительного элемента: номер строки (целое число от 1 до 10000) и номер позиции в строке (целое число от 1 до 10000). Точка экрана, в которую попала хотя бы одна частица, считается светлой, точка, в которую ни одна частица не попала, – тёмной.

Вам необходимо по заданному протоколу определить номер строки с наибольшим количеством светлых точек в нечётных позициях. Если таких строк несколько, укажите номер первой из подходящих строк.

**Входные данные** представлены в файле **26-82.txt** следующим образом. В первой строке входного файла записано целое число N – количество частиц, попавших на экран. В каждой из следующих N строк записаны по два числа, разделённые пробелом: номер строки и номер позиции в строке.

Запишите в ответе два числа: сначала наибольшее количество светлых точек в нечётных позициях одной строки, затем – номер строки, в которой находятся эти точки.

**Пример входного файла:**

**7**

**1 2**

**2 3**

**3 6**

**2 5**

**1 4**

**2 5**

**2 3**

При таких исходных данных в строке 2 имеются две точки в чётных позициях (3 и 5).

1. (**А. Богданов**) В гостинице составляют недельный план уборки номеров после отъезда клиентов. Все номера одинаковые и пронумерованы с 1 до К. В основе плана – журнал заявок, в каждой из которых записано время заезда и время выезда для N заявок. Заявки поступают в случайном порядке. На начало недели все номера подготовлены к заселению. После отъезда клиента на уборку номера отводится 30 минут. Уборка начинается в следующую минуту после освобождения номера. Клиент может заезжать в подготовленный номер в следующую минуту после окончания уборки. Если подготовленных номеров несколько, то выбирается номер **с максимальным временем простоя**; из номеров с одинаковым временем простоя – последний номер. Если подготовленных номеров нет, клиент ждет первый подготовленный номер; при этом время отъезда не меняется. Если первый номер будет готов после запланированного времени отъезда, клиент не ждёт и сразу уезжает.

Определите максимальное время ожидания клиента перед заселением и последний номер, заселенный **в течение этой недели**.

**Входные данные** представлены в файле 26-117.txt следующим образом. В первой строке входных данных задается два числа: K – количество номеров (1 ≤ K ≤ 1000) и N – количество заявок (1 ≤ N ≤ 100000). В каждой из последующих N строк указано время заезда и время выезда в минутах, начиная с 0:00 воскресенья.

Запишите в ответе два числа: максимальное время ожидания клиента перед заселением и последний номер, заселенный **в течение этой недели**.

**Пример входного файла:**

2 5

10 30

15 40

40 65

55 80

56 100

При таких исходных данных первый клиент в минуту 10 сразу заезжает в номер 2, в 15-ю минуту второй клиент заезжает в номер 1 (без ожидания). На 30-й минуте первый клиент выезжает из номера 2 и в этом номере сразу начинается уборка, которая заканчивается на 60-й минуте. Поэтому третий клиент, который хотел заселиться на 40-й минуте, будет ждать 21 минуту и заселится в номер 2 на 61-й минуте. Аналогично четвёртый клиент, который хотел заселиться на 55-й минуте, должен ждать 16 минут, потому что готовый номер 1 будет готов только на 40 + 30 + 1 = 71 минуте. Последний клиент, желающий заселиться на 56-й минуте, фактически сможет сделать это только на 65 + 30 + 1 = 96 минуте, так что он будет ждать 40 минут и заселится в номер 2.

1. (**В. Ланская, Р. Ягафаров**) В некотором городе существует база данных о зарегистрированных автомобильных нарушениях за последние полгода. Марка каждого автомобиля закодирована как целое число, каждый автомобиль отнесен к одному из трёх классов: эконом, средний, премиум. Возможно, что автомобили одной марки имеют разные классы, это указывает на различные варианты комплектации. В таком случае каждую комбинацию «марка – класс» следует рассматривать как отдельную марку. На основе этой статистики необходимо определить водители какой марки автомобилей в сумме заплатили больше всего за все нарушения. Если таких марок несколько, то выбирается марка с наибольшим номером, и если и таких марок несколько, то марка с наивысшим классом.

**Входные данные** представлены в файле 26-155.txt следующим образом. В первой строке входного файла записано натуральное число N, не превышающее 10 000 — количество нарушений, которые занесены в базу. В каждой из следующих N строках через пробел записаны три числа — код марки автомобиля (от 1000 до 9999), сумма штрафа (натуральное число, не превышающее 5000) и код класса (1 — эконом, 2 — средний, 3 — премиум).

Запишите в ответе два числа: код марки с наибольшей суммой штрафов и саму сумму штрафов, которую заплатили водители данной марки автомобиля.

**Пример входного файла:**

4

6666 750 1

2222 1050 2

3333 550 1

2222 1100 2

При таких исходных данных наибольшую сумму (1050 + 1100 = 2150) заплатили водители марки 2222 класса 2.