Московская олимпиада школьников. Информатика. 6 класс. Отборочный этап, 2023/24

15 дек 2023 г., 00:00 — 17 янв 2024 г., 23:59

№ 1

100 баллов

Даша и лимонад

Даша очень любит лимонад! Сегодня она сходила в магазин и купила n бутылок лимонада, каждая из которых объемом t литров.

Она поставила их всех в один ряд и за один шаг делает следующее: берет одну бутылку, затем, если бутылка не пуста, выпивает из нее 1 литр, ставит обратно и переходит к следующей. После последней бутылки она переходит снова к первой.

Изначально все бутылки полные. Даша хочет узнать: сколько бутылок будет выпито полностью, если она сделает k вышеописанных шагов? Помогите ей выяснить это.

Формат входных данных

Первая строка содержит три целых числа n, t и k ($1 \le n \le 10^9, 1 \le t \le 10^9, 1 \le k \le 10^9$) — количество бутылок, объем каждой бутылки и количество шагов соответственно.

Формат выходных данных

Выведите одно число — количество полностью выпитых бутылок.

Критерии оценивания

Номер подзадачи	Баллы	Ограничения	Комментарий
0	0	—	Примеры из условия.
1	40	$1 \leqslant n \leqslant 100$	Баллы начисляются за каждый тест в отдельности.
2	60	Основные ограничения	Баллы начисляются за каждый тест в отдельности.

Примеры

```
2 2 2
```

```
0
```

```
2 1 1
```

```
1
```

```
4 3 10
```

```
2
```

```
4 3 100
```

```
4
```

Ограничения

Время выполнения: 1 секунда

Память: 256 MB

Код C++

```cpp
#include <iostream>
#include <cmath>

using namespace std;

int main() {
        long long n, t, k;

        cin >> n >> t >> k;
    long long a = (t - 1) * n;
    if (k <= a)
        cout << 0;
    else {
      k -= a;
      cout << min(n, k);
    }
        return 0;
}

```

№ 2

100 баллов

Пробка

Глеб оказался в автомобильной пробке. Применив зоркий глаз, Глеб насчитал, что перед ним стоят n машин. Также он знает, что зеленый свет горит ровно a секунд и за каждую секунду зеленого света с пробки успевают уехать ровно b машин. Красный свет горит ровно c секунд.

Глеб хочет узнать, сколько еще секунд он будет стоять в пробке, если прямо сейчас загорится зеленый свет.

Формат входных данных

Первая строка входных данных содержит одно целое число n ($1 \le n \le 10^9$) — количество машин, стоящих перед Глебом.

Вторая строка входных данных содержит три целых числа a, b и c ($1 \le a, b, c \le 10^9$).

Формат выходных данных

Выведите единственное число — сколько секунд придется стоять Глебу в пробке.

Обратите внимание, что ответ в этой задаче может быть довольно большим и не помещаться в 32-битные типы данных. Рекомендуется использовать 64-битный тип данных, например long long в C++, long в Java или int64 в Pascal.

Критерии оценивания

В этой задаче каждый тест, кроме тестов из условия, оценивается независимо.

Примеры

```
5
4 2 3
```

```
3
```

```
11
4 1 3
```

```
18
```

```
2
2 1 5
```

```
8
```

```
2
1 2 5
```

```
7
```

Ограничения

Время выполнения: 1 секунда

Память: 256 MB

Код C++

```cpp
#include <bits/stdc++.h>

using namespace std;
#define int long long

signed main() {
    int n, a, b, c;
    cin >> n >> a >> b >> c;
    int n1 = (n + 1 + b - 1) / b * b;
    int n2 = n1 / b;
    int k1 = (n2 - 1) / a * (a + c);
    cout << k1 + ((n2 - 1) % a + 1);
}

```

№ 3

100 баллов

Преобразование Алисы

Алисе подарили массив a, состоящий из n целых чисел. Она очень любит странные способы преобразования массивов, так что сегодня она придумала следующий алгоритм:

\begin
  \item Ищет три подряд идущих одинаковых числа
  \item Если они нашлись, то она удаляет одно из этих чисел, и переходит к первому шагу, иначе работа алгоритма завершается
\end

Она просит вас узнать, как будет выглядеть массив после исполнения описанного выше алгоритма.

Формат входных данных

Первая строка входных данных содержит одно целое число n ($1 \le n \le 2 \cdot 10^5$) — количество элементов массива.

Вторая строка входных данных содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — элементы массива a.

Формат выходных данных

Выведите массив после исполнения алгоритма.

Критерии оценивания

Номер подзадачи	Баллы	Ограничения	Комментарий
0	0	—	Примеры из условия.
1	40	$1 \leqslant n \leqslant 100$	Баллы начисляются за каждый тест в отдельности.
2	60	Основные ограничения	Баллы начисляются за каждый тест в отдельности.

Примеры

```
6
4 3 10 10 10 1
```

```
4 3 10 10 1
```

```
5
1 1 1 1 1
```

```
1 1
```

```
5
1 2 3 4 9
```

```
1 2 3 4 9
```

Ограничения

Время выполнения: 1 секунда

Память: 256 MB

Код C++

```cpp
#include <bits/stdc++.h>

#define ll long long
#define ld long double
#define fast ios_base::sync_with_stdio(0); cin.tie(0);

using namespace std;

ll INF = (ll) 1e20;
ll mod = (ll) 1e9 + 7;

mt19937
gen(chrono::steady_clock::now().time_since_epoch().count());

signed main() {
    fast
    ll n;
    cin >> n;
    vector<ll> a(n);
    for (int i = 0; i < n; ++i) {
        cin >> a[i];
    }
    vector<ll> ans;
    for (int i = 0; i < n - 2; ++i) {
        if (a[i] != a[i + 1] || a[i] != a[i + 2]) {
            ans.push_back(a[i]);
        }
    }
    if(n > 1)
        ans.push_back(a[n - 2]);
    ans.push_back(a[n - 1]);
    for (auto i : ans) {
        cout << i << ' ';
    }
}
```

№ 4

100 баллов

Очень простая задача

Целое число называется *простым*, если оно не меньше двух и не делится ни на какое целое положительное число, кроме единицы и самого себя.

От вас требуется написать программу, которая ищет отрезок из L последовательных натуральных чисел, содержащий ровно K простых чисел. Чтобы результаты было легче анализировать, вас просят ограничиться в поисках первыми **тридцатью тысячами** натуральных чисел.

Формат входных данных

На вход программе подаются целые числа L и K ($1 \leqslant L \leqslant 30\,000, 0 \leqslant K \leqslant L$), каждое в отдельной строке.

Формат выходных данных

Если в пределах до $30\,000$ найдётся отрезок из L подряд идущих натуральных чисел, среди которых ровно K простых, выведите минимальное и максимальное число на этом отрезке. В противном случае выведите единственное число -1. Если существует несколько отрезков, удовлетворяющих условию, выведите любой.

Критерии оценивания

Все тесты оцениваются независимо. Разные тесты могут оцениваться в разное количество баллов.

Примеры

```
20
5
```

```
8 27
```

```
100
66
```

```
-1
```

Ограничения

Время выполнения: 1 секунда

Память: 256 MB

Код C++

```cpp
#include <cstdio>
#include <iostream>
using namespace std;

#define MAX_PRIME 30000

bool is_prime(int x)
{
        if (x < 2) {
                return false;
        }
        for (int i = 2; i * i <= x; ++i) {
                if (x % i == 0) {
                        return false;
                }
        }
        return true;
}

int main()
{
        //freopen("prime-segment.in","r",stdin);
        //freopen("prime-segment.out","w",stdout);
        int len, k;
        cin >> len >> k;

        int cnt = 0;
        for (int i = 0; i < len; ++i) {
                cnt += is_prime(i);
        }

        for (int x = 1; x + len - 1 <= MAX_PRIME; ++x) {
                cnt += is_prime(x + len - 1) - is_prime(x - 1);
                if (cnt == k) {
                        cout << x << " " << x + len - 1 << endl;
                        return 0;
                }
        }
        cout << "-1\n";
        return 0;
}

```

№ 5

100 баллов

Пёстрая дата

По текущей дате определите ближайшую следующую дату, запись которой в виде день, месяц и год состоит из различных цифр.

Учтите, что если день или номер месяца меньше 10, то они записываются без лишних нулей. Например, дата «8 марта 2019 года» запишется так: «8.3.2019».

Гарантируется, что изначальная дата корректна.

Формат входных данных

В первой строке задано одно целое число d ($1 \le d \le 31$) — день текущей даты.

Во второй строке задано одно целое число m ($1 \le m \le 12$) — месяц текущей даты.

В третьей строке задано одно целое число y ($1000 \le y \le 10^6$) — год текущей даты.

Формат выходных данных

Выведите одну строку — ближайшую следующую дату, в записи которой все цифры различны.

Критерии оценивания

Все тесты в этой задаче оцениваются независимо.

Примечание

Високосный год — такой год, что в нём 366 дней, а именно добавляется 29 февраля. Год является високосным, если он делится на 400 или делится на 4, но не делится на 100.

Примеры

```
18
12
2018
```

```
4.3.2019
```

```
1
2
3456
```

```
7.2.3456
```

Ограничения

Время выполнения: 1 секунда

Память: 256 MB

Код C++

```cpp
#include <fstream>
#include <iostream>
using namespace std;

bool leap_year(int y) {
	return (y % 4 == 0 && y % 100 != 0 || y % 400 == 0);
}

int day_in_month(int m, int y) {
	if (m == 2)
		if (leap_year(y))
			return 29;
		else
			return 28;
	else if (m == 4 || m == 6 || m == 9 || m == 11)
		return 30;
	else
		return 31;
}

void inc(int & d, int & m, int & y) {
	d += 1;
	int dm = day_in_month(m, y);
	if (d > dm) {
		d = 1;
		m += 1;
		if (m > 12) {
			m = 1;
			y += 1;
		}
	}
}

bool good_data(int d, int m, int y) {
	int a[10];
	for (int i = 0; i < 10; i++) a[i] = 0;
	a[d % 10] += 1;
	if (d > 9)	a[d / 10] += 1;
	a[m % 10] += 1;
	if (m > 9) a[m / 10] += 1;
	while (y > 0) {
		a[y % 10] += 1;
		y /= 10;
	}
	bool good = true;
	for (int elem : a)
		if (elem > 1) {
			good = false;
			break;
		}
	return good;
}
int main() {
	int day, month, year;

	cin >> day >> month >> year;
	inc(day, month, year);
	while (!good_data(day, month, year))
		inc(day, month, year);
	cout << day << '.' << month << '.' << year;

	return 0;
```

```
63  }
64
```

№ 6

100 баллов

Закрась по правилам

Дан клетчатый прямоугольник из N строк по M клеток, каждая из которых или уже закрашена, или еще не закрашена. Если в каком-либо квадрате размером 2×2 три клетки уже закрашены, то можно закрасить и четвёртую клетку.

Оцените, сколько клеток могут в итоге оказаться закрашенными.

Формат входных данных

Первая строка входных данных содержит два целых числа N и M ($1 \leqslant N, M \leqslant 500$) — количество строк и столбцов в прямоугольнике. Следующие N строк по M символов описывают клетки прямоугольника. Символ «.» соответствует незакрашенной клетке, а «#» — закрашенной клетке. Строки нумеруются от 1 до N, столбцы — от 1 до M.

Формат выходных данных

Выведите одно число — максимальное количество клеток, которые могут оказаться закрашенными.

Критерии оценивания

Каждый тест оценивается независимо. Решения, правильно работающие для ограничений $1 \leqslant N, M \leqslant 50$, будут набирать не менее 60 баллов.

Примеры

```
2 2
##
#.
```

```
4
```

```
3 4
#...
#...
###.
```

```
9
```

```
3 5
...##
#....
#.#..
```

```
5
```

Ограничения

Время выполнения: 2 секунды

Память: 256 MB

Код C++

```cpp
#include <iostream>
#include <vector>
#include <cstdio>

using namespace std;

int table[1000][1000], n, m;
vector < pair < int, int> > st;

int sumSquare(int i, int j)
{
    return table[i][j] + table[i + 1][j] + table[i][j + 1] +
table[i + 1][j + 1];
}
void pushSquare(int i, int j)
{
    if ((0 <= i) && (i < n - 1) && (0 <= j) && (j < m - 1))
        if (sumSquare(i, j) == 3)
            st.push_back(make_pair(i, j));
}

void pushNeighbors(int i, int j)
{
    pushSquare(i - 1, j - 1);
    pushSquare(i, j - 1);
    pushSquare(i - 1, j);
    pushSquare(i, j);
}
void updTile(int i, int j)
{
    if ((0 <= i) && (i < n) && (0 <= j) && (j < m))
        if (!table[i][j])
            table[i][j] = 1;
            pushNeighbors(i, j);
}

void updSquare(int i, int j)
{
    updTile(i, j);
    updTile(i + 1, j);
    updTile(i, j + 1);
    updTile(i + 1, j + 1);
}

int main()
{
   // freopen("f.in", "r", stdin);
  //    freopen("f.out", "w", stdout);
    cin >> n >> m;
    char c;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cin >> c;
            if (c == '#')
                table[i][j] = 1;
        }
    }
    for (int i = 0; i < n - 1; i++)
        for (int j = 0; j < m - 1; j++)
            if (sumSquare(i, j) == 3)
```

```
                st.push_back(make_pair(i, j));
    while (!st.empty())
    {
        pair < int, int > p = st.back();
        st.pop_back();
        if (sumSquare(p.first, p.second))
            updSquare(p.first, p.second);
    }
    int answer = 0;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            answer += table[i][j];
    cout << answer << endl;
    return 0;
}

```