**Задача 1. Хеш-таблица**

Реализуйте структуру данных типа “множество строк” на основе динамической хеш-таблицы с открытой адресацией. Хранимые строки состоят из строчных латинских букв.

Структура данных должна поддерживать операции добавления строки в множество, удаления строки из множества и проверки принадлежности данной строки множеству.

**1\_1.** Для разрешения коллизий используйте квадратичное пробирование. i-ая проба  
g(k, i)=g(k, i-1) + i (mod m). - степень двойки.

**1\_2.** Для разрешения коллизий используйте двойное хеширование.

Формат входных данных

Каждая строка входных данных задает одну операцию над множеством. Запись операции состоит из типа операции и следующей за ним через пробел строки, над которой проводится операция.

Тип операции – один из трех символов:  
 + означает добавление данной строки в множество;   
 - означает удаление строки из множества;   
 ? означает проверку принадлежности данной строки множеству.

При добавлении элемента в множество НЕ ГАРАНТИРУЕТСЯ, что он отсутствует в этом множестве. При удалении элемента из множества НЕ ГАРАНТИРУЕТСЯ, что он присутствует в этом множестве.

Формат выходных данных

Программа должна вывести для каждой операции одну из двух строк OK или FAIL, в зависимости от того, встречается ли данное слово в нашем множестве.

|  |  |
| --- | --- |
| stdin | stdout |
| + hello + bye ? bye  + bye - bye ? bye ? hello | OK  OK  OK  FAIL  OK  FAIL  OK |

**Задача 2. Динамика с одним параметром.**Предназначена для решения на семинаре 14 декабря.

**2\_1. Дорожки.** В тридевятом царстве все деревни расположены на одной прямой в труднопроходимом лесу. Чтобы жители не скучали по вечерам, местный царь решил проложить дорожки между некоторыми деревнями. Царь хочет, чтобы каждая деревня была соединена хотя бы с одной другой, и также желает сэкономить. Определите минимальную сумму длин дорожек, необходимую для реализации царской воли.

Формат входных данных.

Вначале задается n - число деревень, затем n координат.

Формат выходных данных.

Вывести число - минимальную сумму длин дорожек.

|  |  |
| --- | --- |
| stdin | stdout |
| 6 6 12 4 13 3 14 | 5 |

**2\_2. Телефонизация в год лошади.** В наступающий год лошади телефонная компания решила всем новым абонентам выдавать “красивые” телефонные номера. Номер считается красивым если его можно набрать, перемещаясь по циферблату телефона ходом шахматного коня.

Циферблат выглядит следующим образом:

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
|  | 0 |  |

Телефонный номер не может начанаться с “0” и “8”.

Например “красивым” 7-значным номером считается 1616161.

Требуется посчитать какое максимальное количество абонентов сможет привлечь компания.

Формат входных данных:

На вход подается длина номера N.

Формат выходных данных:

Требуется вывести количество N-значных комбинаций, удовлетворяющих условию задачи.

|  |  |
| --- | --- |
| stdin | stdout |
| 1 | 8 |
| 2 | 16 |

Задача 3. Динамика с двумя параметрами

Дано N кубиков. Требуется определить каким количеством способов можно выстроить из этих кубиков пирамиду.

Формат входных данных:

На вход подается количество кубиков N.

Формат выходных данных:

Вывести число различных пирамид из N кубиков.

Критерии, которым должна удовлетворять пирамида, отличаются по вариантам:

**3\_1.** **Высокая пирамида.** Каждый вышележащий слой пирамиды должен быть не больше нижележащего.

|  |  |
| --- | --- |
| stdin | stdout |
| 3 | 3 |
| 5 | 7 |
| 7 | 15 |

**3\_2. Широкая пирамида.** Каждый вышележащий слой пирамиды должен быть строго меньше нижележащего.

|  |  |
| --- | --- |
| stdin | stdout |
| 3 | 2 |
| 5 | 3 |
| 7 | 5 |

Задача 4. Жадные алгоритмы

**4\_1.** **Атлеты.**

В город *N* приехал цирк с комндой атлетов. Они хотят удивить горожан города *N* — выстроить из своих тел башню максимальной высоты. Башня — это цепочка атлетов, первый стоит на земле, второй стоин у него на плечах, третий стоит на плечах у второго и т.д.

Каждый атлет характеризуется силой *si* (kg) и массой *mi* (kg). Сила — это максимальная масса, которую атлет способен держать у себя на плечах.

К сожалению ни один из атлетов не умеет программировать, так как всю жизнь они занимались физической подготовкой, и у них не было времени на изучение языков программирования. Помогите им, напишите программу, которая определит максимальную высоту башни, которую они могут составить.

Известно, что если атлет тяжелее, то он и сильнее:

если *mi*>*mj*, то *si* > *sj*.

Атлеты равной массы могут иметь различную силу.

Формат входных данных:

Вход содержит только пары целых чисел — массу и силу атлетов. Число атлетов 1 ≤ *n* ≤ 100000. Масса и сила являются положительными целыми числами меньше, чем 2000000.

Формат выходных данных:

Выход должен содержать натуральное число — максимальную высоту башни.

|  |  |
| --- | --- |
| stdin | stdout |
| 3 4 2 2 7 6 4 5 | 3 |

**4\_2. Заявки на переговоры.**

В большой IT-фирме есть только одна переговорная комната. Желающие посовещаться заполняют заявки с желаемым временем начала и конца. Ваша задача определить максимальное количество заявок, которое может быть удовлетворено.

Формат входных данных:

Вход содержит только пары целых чисел — начала и концы заявок.

Формат выходных данных:

Выход должен содержать натуральное число — максимальное число заявок.

|  |  |
| --- | --- |
| stdin | stdout |
| 1 2  2 5  7 10  2 3  6 7  4 7  3 6  3 5  1 2  4 5 | 5 |

Задача 5. Код Хаффмана