

Задание по программированию: Ежемесячные дела

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

Deadline The assignment was due on 29 сент. г., 23:59 PDT
You can still pass this assignment before the course ends.

Instructions My submission

У каждого из нас есть ежемесячные дела, каждое из которых нужно выполнять в конкретный день каждого месяца: оплата счетов за электричество, абонентская плата за связь и пр. Вам нужно реализовать работу со списком таких дел, а именно, обработку следующих операций:

ADD i s

Добавить дело с названием s в день i .

NEXT

Закончить текущий месяц и начать новый. Если новый месяц имеет больше дней, чем текущий, добавленные дни изначально не будут содержать дел. Если же в новом месяце меньше дней, дела со всех удаляемых дней необходимо будет переместить на последний день нового месяца.

Обратите внимание, что количество команд этого типа может превышать 11.

DUMP i

Вывести все дела в день i .

Изначально текущим месяцем считается январь. Количества дней в месяцах соответствуют Григорианскому календарю с той лишь разницей, что в феврале всегда 28 дней.

Формат ввода

Сначала число операций Q , затем описания операций.

Названия дел s уникальны и состоят только из латинских букв, цифр и символов подчёркивания. Номера дней i являются целыми числами и нумеруются от 1 до размера текущего месяца.

Формат вывода

Для каждой операции типа **DUMP** в отдельной строке выведите количество дел в соответствующий день, а затем их названия, разделяя их пробелом. Порядок вывода дел в рамках каждой операции значения не имеет.

Пример

Ввод

```
1 12
2 ADD 5 Salary
3 ADD 31 Walk
4 ADD 30 WalkPreparations
5 NEXT
6 DUMP 5
7 DUMP 28
8 NEXT
9 DUMP 31
10 DUMP 30
11 DUMP 28
12 ADD 28 Payment
13 DUMP 28
14
```

Вывод

```
1 1 Salary
2 2 WalkPreparations Walk
3 0
4 0
5 2 WalkPreparations Walk
6 3 WalkPreparations Walk Payment
7
```

Указание

Для дописывания всех элементов вектора v2 в конец вектора v1 удобно использовать метод insert:

```
1 v1.insert(end(v1), begin(v2), end(v2));
```

Кроме того, элементом вектора может быть любой тип, в том числе и другой вектор. Например, `vector<vector<int>>` — это вектор, элементами которого являются вектора целых чисел (то есть двумерный массив). Пример использования:

```
1 vector<vector<int>> m(10); // Создаём вектор из десяти векторов целых
   чисел
2 m[0].push_back(5); // Добавляем элементы в первый вектор
3 m[0].push_back(15);
4 cout << m[0][1]; // Выведет 15 — второй элемент первого вектора
5
6 m[1].push_back(3);
7 for (int x : m[1]) {
8     // Перебираем все элементы второго
9 }
```

Задание по программированию: Автобусные остановки — 2

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

Deadline The assignment was due on 29 сент. г., 23:59 PDT
You can still pass this assignment before the course ends.

Instructions My submission

В этой задаче вам нужно присваивать номера автобусным маршрутам.

А именно, для каждого маршрута, заданного набором названий остановок, нужно либо выдать новый номер (первому маршруту — 1, второму — 2 и т. д.), либо вернуть номер существующего маршрута, которому соответствует такой набор остановок.

Наборы остановок, полученные друг из друга перестановкой остановок, считаются различными (см. пример).

How to s
When you're re
you can upload
part of the assi
"My submission"

Указание

В C++ ключом словаря может быть не только число или строка, но и другой контейнер, например, `vector`.

Формат ввода

Сначала вводится количество запросов Q , затем Q описаний запросов.

Каждый запрос представляет собой положительное количество остановок N , за которым следуют разделённые пробелом N различных названий остановок соответствующего маршрута. Названия остановок состоят лишь из латинских букв и символов подчёркивания.

Формат вывода

Выведите ответ на каждый запрос в отдельной строке.

Если маршрут с данным набором остановок уже существует, в ответ на соответствующий запрос выведите **Already exists for i** , где i — номер маршрута с таким набором остановок. В противном случае нужно выделить введённому набору остановок новый номер i и вывести его в формате **New bus i** .

Пример

Ввод

```
1 4
2 2 Marushkino Kokoshkino
3 1 Kokoshkino
4 2 Marushkino Kokoshkino
5 2 Kokoshkino Marushkino
6
```

Вывод

```
1 New bus 1
2 New bus 2
3 Already exists for 1
4 New bus 3
5
```

Замечание

Будьте внимательны при обновлении словаря: если одновременно в одном выражении вычислять размер словаря и добавлять в него новый элемент, результат может отличаться от ожидаемого на единицу. Чтобы этого избежать, используйте временную переменную.

Например, вместо кода

```
1 map<string, int> m;
2 // ...
3 m["two"] = m.size();
4 // что нужно сделать раньше: вычислить size или добавить новый ключ?
5
```

стоит написать такой код:

```
1 map<string, int> m;
2 // ...
3 int new_value = m.size();
4 m["two"] = new_value;
5 // теперь всё однозначно: нужен размер до добавления элемента
6
```

Задание по программированию: Синонимы

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

Deadline The assignment was due on 29 сент. г., 23:59 PDT
You can still pass this assignment before the course ends.

Instructions My submission

Два слова называются *синонимами* друг друга, если они имеют похожие значения. Реализуйте следующие операции над словарём синонимов:

- **ADD *word1 word2*** — добавить в словарь пару синонимов (*word1*, *word2*).
- **COUNT *word*** — узнать количество синонимов слова *word*.
- **CHECK *word1 word2*** — проверить, являются ли слова *word1* и *word2* синонимами. Слова *word1* и *word2* считаются синонимами, если среди запросов **ADD** был хотя бы один запрос **ADD *word1 word2*** или **ADD *word2 word1***.

Формат ввода

Сначала вводится количество запросов Q , затем Q строк с описаниями запросов. Гарантируется, что в каждом запросе **CHECK** и **ADD** слова *word1* и *word2* различны. Все слова состоят лишь из латинских букв, цифр и символов подчёркивания.

Формат вывода

Для каждого запроса в соответствующей строке выведите ответ на него:

- В ответ на запрос **COUNT *word*** выведите единственное целое число — количество синонимов слова *word*.
- В ответ на запрос **CHECK *word1 word2*** выведите строку **YES**, если *word1* и *word2* являются синонимами, и **NO** в противном случае.

Пример

Ввод

```
1 8
2 ADD program code
3 COUNT cipher
4 ADD code cipher
5 COUNT code
6 COUNT program
7 CHECK code program
8 CHECK program cipher
9 CHECK cpp java
10
```

Вывод

```
1 0
2 2
3 1
4 YES
5 NO
6 NO
7
```

4

Задание по программированию: Имена и фамилии — 1

Вы не отправили работу. Для успешной сдачи вам необходимо набрать 1/1 баллов.

Deadline The assignment was due on 6 окт. г., 23:59 PDT
You can still pass this assignment before the course ends.

Instructions My submission

Реализуйте класс для человека, поддерживающий историю изменений человеком своих фамилии и имени.

Nov

```
1 class Person {
2 public:
3 void ChangeFirstName(int year, const string& first_name) {
4 // добавить факт изменения имени на first_name в год year
5 }
6 void ChangeLastName(int year, const string& last_name) {
7 // добавить факт изменения фамилии на last_name в год year
8 }
9 string GetFullName(int year) {
10 // получить имя и фамилию по состоянию на конец года year
11 }
12 private:
13 // приватные поля
14 };
15
```

Считайте, что в каждый год может произойти не более одного изменения фамилии и не более одного изменения имени. При этом с течением времени могут открываться всё новые факты из прошлого человека, поэтому годá в последовательных вызовах методов `ChangeLastName` и `ChangeFirstName` не обязаны возрастать.

Гарантируется, что все имена и фамилии непусты.

Строка, возвращаемая методом `GetFullName`, должна содержать разделённые одним пробелом имя и фамилию человека по состоянию на конец данного года.

- Если к данному году не случилось ни одного изменения фамилии и имени, верните строку **"Incognito"**.
- Если к данному году случилось изменение фамилии, но не было ни одного изменения имени, верните **"last_name with unknown first name"**.
- Если к данному году случилось изменение имени, но не было ни одного изменения фамилии, верните **"first_name with unknown last name"**.

Пример

Код

```
1 int main() {
2     Person person;
3
4     person.ChangeFirstName(1965, "Polina");
5     person.ChangeLastName(1967, "Sergeeva");
6     for (int year : {1900, 1965, 1990}) {
7         cout << person.GetFullName(year) << endl;
8     }
9
10    person.ChangeFirstName(1970, "Appolinaria");
11    for (int year : {1969, 1970}) {
12        cout << person.GetFullName(year) << endl;
13    }
14
15    person.ChangeLastName(1968, "Volkova");
16    for (int year : {1969, 1970}) {
17        cout << person.GetFullName(year) << endl;
18    }
19
20    return 0;
21 }
22
```

Вывод

```
1 Incognito
2 Polina with unknown last name
3 Polina Sergeeva
4 Polina Sergeeva
5 Appolinaria Sergeeva
6 Polina Volkova
7 Appolinaria Volkova
8
```

5

Класс `Rational` (делала по примеру с 3-го семестра)