

## Programming Assignment: Ежемесячные дела

✔ Passed · 1/1 points

**Deadline** Pass this assignment by Mar 16, 9:59 AM MSK

**Instructions** My submission

Discussions

У каждого из нас есть повторяющиеся 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 на месяц, а именно, реализовать набор следующих операций:

### How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.

#### ADD *i* *s*

Назначить дело с названием *s* на день *i* текущего месяца.

#### DUMP *i*

Вывести все дела, запланированные на день *i* текущего месяца.

#### NEXT

Перейти к списку дел на новый месяц. При выполнении данной команды вместо текущего (старого) списка дел на текущий месяц создаётся и становится активным (новый) список дел на следующий месяц: все дела со старого списка дел копируются в новый список. После выполнения данной команды новый список дел и следующий месяц становятся текущими, а работа со старым списком дел прекращается. При переходе к новому месяцу необходимо обратить внимание на разное количество дней в месяцах:

- если следующий месяц имеет больше дней, чем текущий, «дополнительные» дни необходимо оставить пустыми (не содержащими дел);
- если следующий месяц имеет меньше дней, чем текущий, дела со всех «лишних» дней необходимо переместить на последний день следующего месяца.

### Замечания

- Историю списков дел хранить **не требуется**, работа ведётся только с текущим списком дел текущего месяца. Более того, при создании списка дел на следующий месяц, он **«перетирает»** предыдущий список.
- Обратите внимание, что количество команд **NEXT** в общей последовательности команд при работе со списком дел может превышать 11.
- Начальным 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
2 // целых чисел
3 m[0].push_back(5); // Добавляем элементы в первый вектор
4 m[0].push_back(15);
5 cout << m[0][1]; // Выведет 15 – второй элемент первого вектора
6
7 m[1].push_back(3);
8 for (int x : m[1]) {
9     // Перебираем все элементы второго
```