Вам предлагается выступить в качестве процессора и преобразовать несколько логических адресов в физические. Формат входных данных следующий:

* в первой строке вам даны 3 числа ***m,q,r≥0*** , где ***q*** - это количество запросов, на которые вам нужно ответить, ***r*** - физический адрес корневой таблицы страниц
* следующих ***m*** строках записаны пары ***paddr*** и ***value*** - описание физической памяти, каждая пара значит, что по физическому адресу ***paddr***хранится 64 битное значение ***value***, при этом гарантируется, что все ***paddr*** различны, выровнены на границу 8 байт и помещаются в 64 бита
* в последних ***q*** строках идут целые числа - логические адреса, которые вам нужно преобразовать в физические, для каждого из этих чисел нужно вывести на отдельной строке либо физический адрес, либо слово "***fault***", если преобразовать логический адрес в физический нельзя.

Считайте, что таблица страниц имеет формат 64 битного режима x86 (4 уровня, каждая страница 4 KB, каждая запись 8 байт, формат записи был показан в презентации), но вы можете игнорировать все поля, кроме бита присутствия (на картинке бит **P** - нулевой бит) и собственно физического адреса.

Для всех физических адресов, не указанных во входных данных (среди ***m*** пар ***paddr value***), считайте, что по этим адресам хранятся нули.

**ВАЖНО:** это было неочевидно из видео, но все физические адреса, которые хранятся в записях таблицы страниц должны быть **выровнены, как минимум, на границу 4096 байт (4Kb), т. е. младшие 12 бит физических адресов всегда равны 0**, соответственно, хранить младшие биты нет смысла и в записе таблицы страниц они не хранятся - их место занимают специальные флаги. Убедитесь, что вы понимаете приведенный пример.

**ВАЖНО2:** после каждой неверной попытки вам требуется скачать новый набор данных и использовать его для следующей попытки.

**Sample Input:**

4 4 0

0 4097

4096 8193

8192 12289

12288 16385

0

4096

42

131313

**Sample Output:**

16384

fault

16426

fault