**Таргетинг на погоду**

Осень уже в самом разгаре --- а это значит, что нас всех ждут и дождливые, и тёплые дни. В дождь людям нужны зонты и дождевики, и производителям этих товаров крайне важно отслеживать рост интереса к ним. С другой стороны, производителям прохладительных напитков важны тёплые дни, когда на продвижение можно потратить бóльшие бюджеты.

Итак, владельцам рекламных объявлений определённых тематик требуется возможность управлять настройками показов в зависимости от погоды. В этой задаче вам потребуется реализовать простейшую версию этой функциональности --- таргетинг *на температуру* в определённые диапазоны дней.

В вашем распоряжении есть исторические данные о температуре, а также прогноз на ближайшее будущее. А именно:

* температура сегодня (этот день имеет номер 00);
* температура за прошедшие �*N* дней (вчерашний имеет номер −1−1, позавчерашний --- −2−2 и так далее до дня −�−*N*, самого раннего из известных);
* прогноз температуры на ближайшие �*N* дней (завтрашний имеет номер 11, послезавтрашний --- 22 и так далее до дня �*N*).

Кроме того, есть данные о настройках показа �*B* рекламных объявлений. Каждому из них соответствуют четыре числа ����*tmin*​, ����*tmax*​, ����*dmin*​, ����*dmax*​, означающие, что объявление можно показывать только в том случае, если за период от дня ����*dmin*​ до дня ����*dmax*​ включительно температура принимала или примет значение от ����*tmin*​ до ����*tmax*​ включительно. При этом гарантируется, что ����≤0≤����*dmin*​≤0≤*dmax*​, то есть диапазон дней должен включать сегодняшний.

Обратите внимание, что **температура не дискретна!** Например, если вчера значение температуры было −5−5, а сегодня −2−2, то необходимо считать, что в период, включающий сегодняшний и вчерашний дни, температура принимала не только значения −5−5 и −2−2, но и −4−4 и −3−3 (и даже −3,14−3,14, но в этой задаче все входные данные целочисленны).

Для каждого рекламного объявления определите, может ли оно быть показано в текущий день с учётом настроек и известных погодных данных.

**Формат ввода**

В первой строке входного файла вводится единственное целое неотрицательное число �*N* (�≤100000*N*≤100000).

В следующей строке вводится 2�+12*N*+1 целых чисел --- значения температуры в дни −�−*N*, −�+1−*N*+1, ..., −1−1, 00, 11, ..., �*N*. Значения температуры по модулю не превосходят 109109.

В третьей строке содержится единственное целое положительное число �*B* (�≤100000*B*≤100000) --- количество рекламных объявлений.

Наконец, в каждой из следующих �*B* строк содержатся 4 целых числа ����*tmin*​, ����*tmax*​, ����*dmin*​, ����*dmax*​, описывающие настройки соответствующего объявления. Гарантируется, что −109≤����≤����≤109−109≤*tmin*​≤*tmax*​≤109 и −�≤����≤0≤����≤�−*N*≤*dmin*​≤0≤*dmax*​≤*N*.

**Формат вывода**

Для каждого из объявлений выведите в отдельной строке «yes», если оно может быть показано согласно настройкам, и «no» в противном случае.

**Пример**

Ввод

2

2 4 5 -2 2

4

-1 -1 -1 1

0 10 -1 0

6 8 -2 2

3 3 -1 0

Вывод

yes

yes

no

no

**Примечание**

В задаче большой объем входных и выходных данных, для решения на **Java** используйте буферизированный ввод. Не используйте для чтения класс Scanner. Для вывода не используйте System.out.printf(), String.format(), используйте System.out.println() или OutputStreamWriter. Для буферизированного вывода не забывайте вызывать OutputStreamWriter.flush() перед выходом из приложения.

Ограничение памяти

256.0 Мб

Ограничение времени

3 с

Ввод

стандартный ввод или input.txt

Вывод

стандартный вывод или output.txt