# Adwokat

Dostępna pamięć: 256 MB.

Adwokat Bajtazar, współwłaściciel kancelarii *Bajtazar i wspólnicy*, jest jednym z najbardziej rozchwytywanych członków bajtockiej palestry. Nic więc dziwnego, że jest wiecznie zajęty. Każdego dnia umawia się na liczne spotkania i dawno już przestał kontrolować to, czy będzie w stanie uczestniczyć w nich wszystkich. Zatrudnił więc sekretarza, który ma mu pomóc w ogarnięciu tego chaosu. Bajtazar postanowił, że każdego dnia uda się tylko na dwa spotkania i będzie w nich uczestniczył od początku do końca. Na pozostałe spotkania zostaną oddelegowani asystenci, których w kancelarii Bajtazara nie brakuje.

Niestety, czasem w przepełnionym kalendarzu Bajtazara trudno znaleźć dwa spotkania, których terminy nie nachodzą na siebie. Przyjmujemy, że dwa spotkania nie nachodzą na siebie, jeśli jedno z nich zaczyna się *ściśle* po zakończeniu drugiego. Pomóż sekretarzowi Bajtazara i napisz program, który poradzi sobie z tym problemem.

## Wejście

Pierwszy wiersz wejścia zawiera dwie liczby całkowite $n$ i $m$ ($2 \leqslant n \leqslant 500\,000$, $1 \leqslant m \leqslant 20$), oznaczające liczbę spotkań w terminarzu Bajtazara oraz liczbę dni, które obejmuje terminarz.

W każdym z kolejnych $n$ wierszy opisane jest jedno spotkanie. Opis spotkania składa się z trzech liczb całkowitych $a_i, b_i, d_i$ ($1 \leqslant a_i < b_i \leqslant 80\,000\,000$, $1 \leqslant d_i \leqslant m$), które oznaczają, że dnia $d_i$ Bajtazar ma zaplanowane spotkanie, które rozpocznie się dokładnie $a_i$ milisekund po północy i zakończy się $b_i$ milisekund po północy.

## Wyjście

Twój program powinien wypisać na wyjście $m$ wierszy. W $i$-tym z tych wierszy powinna znaleźć się informacja, czy Bajtazar może uczestniczyć w dwóch spotkaniach $i$-tego dnia. Jeśli jest to niemożliwe, należy wypisać jedno słowo `NIE`. W przeciwnym razie należy wypisać `TAK`, a następnie numery dwóch spotkań, w których może uczestniczyć Bajtazar. Spotkania są ponumerowane od 1 do $n$, zgodnie z ich kolejnością na wejściu. Pierwsze z tych dwóch spotkań powinno się rozpoczynać wcześniej. Drugie spotkanie powinno zacząć się co najmniej milisekundę po zakończeniu pierwszego.

Jeśli istnieje wiele poprawnych odpowiedzi, Twój program powinien wypisać dowolną z nich.

## Przykład

Dla danych wejściowych:

```
6 3
3 5 1
2 4 2
1 8 1
6 7 3
3 5 2
7 12 1
```

poprawnym wynikiem jest:

```
TAK 1 6
NIE
NIE
```