Rádiové věže

V Jakartě je N rádiových věží. Všechny věže leží na jedné přímce a jsou očíslované od 0 do $N-1$ v pořadí zleva doprava. Věž i (kde $0 \le i \le N-1$) má výšku $H[i]$ a navíc jsou výšky věží **po dvou různé**, tedy $H[i] \neq H[j]$ pro $i \neq j$.

Interferenční hodnota δ je kladné celé číslo. Dvojice věží i a j (kde $0 \le i \le j \le N-1$ spolu může komunikovat právě tehdy, když existuje věž k (věži k budeme říkat *prostředník*) splňující obě následující podmínky:

- Věž i je nalevo od věže k a věž j je napravo od věže k, tedy $i < k < j$.
- Věže i i j mají obě výšku nejvýše $H[k] - \delta$.

Pak Dengklek si chce pronajmout několik rádiových věží pro svoji novou stanici. Vaším úkolem odpovědět na Q otázek. Otázky jsou následujícího tvaru: kolik nejvýše věží si *Pak Dengklek* může pronajmout, má-li dané L, R a D ($0 \le L \le R \le N-1$ a $D > 0$) a musí splnit následující podmínky:

- *Pak Dengklek* si může pronajmout pouze stanice s indexem mezi L a R (včetně), tedy s indexem i splňujícím $L \le i \le R$.
- Interferenční hodnota δ je rovna D.
- Všechny dvojice pronajatých věží spolu mohou komunikovat.

Pronajaté věže i a j spolu mohou komunikovat za pomoci věže k jako prostředníka i když si *Pak Dengklek* vež k nepronajme.

Implementační detaily

Implementujte následující funkce:

```
void init(int N, int[] H)
```

- N: počet rádiových věží.
- H: pole délky N obsahující výšky věží.
- Tato funkce bude zavolána právě jednou, a to před prvním voláním funkce `max_towers`.

```
int max_towers(int L, int R, int D)
```

- L, R: hranice úseku, ve kterém může *Pak Dengklek* věže pronajmout.
- D: hodnota δ.
- Tato funkce by měla frátit vrátit maximální počet věží, které si Pak Dengklek může pronajmout pokud může pronajímat pouze věže mezi L a R (včetně) a hodnota δ je D.
- Tato funkce bude volána právě Q-krát

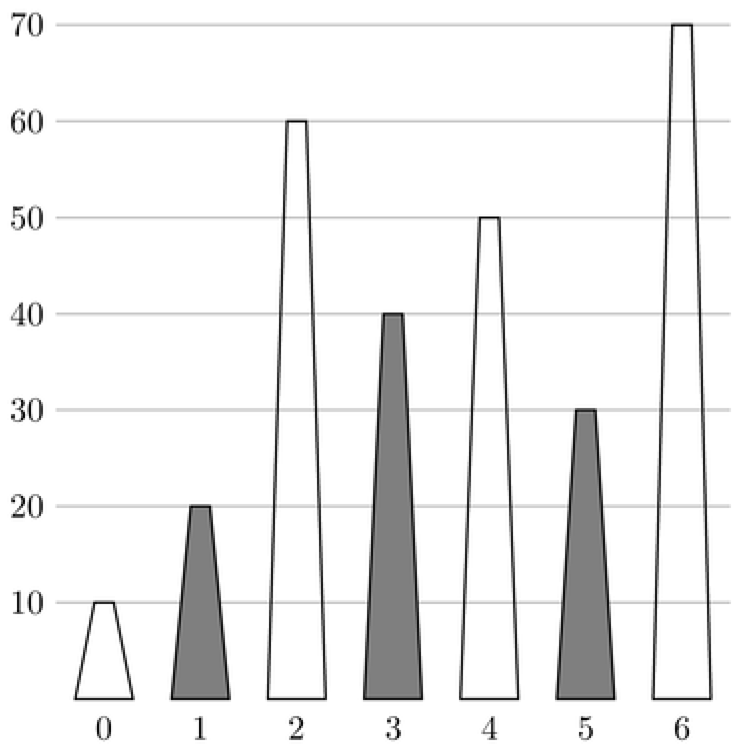
Příklady

Uvažme následující posloupnost volání funkcí:

```
init(7, [10, 20, 60, 40, 50, 30, 70])
```

```
max_towers(1, 5, 10)
```

Pak Dengklek si může pronajmout věže 1, 3, a 5. Tento přilkad je ilustrován na následujícím obrázku, kde šedivé lichoběžníky značí pronajaté věže.



Věže 3 a 5 mohou komunikovat za pomoci věže 4 jako prostředníka, protože $40 \leq 50 - 10$ a $30 \leq 50 - 10$. Věže 1 a 3 mohou komunikovat za pomoci věže 2 jako prostředníka. Věže 1 a 5 mohou komunikovat za pomoci věže 3 jako prostředníka. Není možné si pronajmout více než 3 věže, takže funkce by měla vrátit 3.

```
max_towers(2, 2, 100)
```

V zadaném rozsahu je jenom jedna věž, takže *Pak Dengklek* si může pronajmout nejvýše jednu věž. Funkce by tedy měla vrátit 1.

```
max_towers(0, 6, 17)
```

Pak Dengklek si může pronajmout věže 1 a 3. Věže 1 a 3 mohou komunikovat za pomoci věže 2 jako prostředníka, protože $20 \leq 60 - 17$ a $40 \leq 60 - 17$. Není možné si pronajmout více než 2 věže, takže funkce by měla vrátit 2.

Omezení

- $1 \leq N \leq 100\ 000$
- $1 \leq Q \leq 100\ 000$
- $1 \leq H[i] \leq 10^9$ (pro každé i splňující $0 \leq i \leq N-1$)
- $H[i] \neq H[j]$ (pro každé i a j splňující $0 \leq i < j \leq N-1$)
- $0 \leq L \leq R \leq N-1$
- $1 \leq D \leq 10^9$

Podúlohy

1. (4 body) Existuje k ($0 \leq k \leq N-1$) takové, že $H[i] < H[i+1]$ (pro každé i splňující $0 \leq i \leq k-1$) a $H[i] > H[i+1]$ (pro každé i splňující $k \leq i \leq N-2$).
2. (11 bodů) $Q = 1$, $N \leq 2000$
3. (12 bodů) $Q = 1$
4. (14 bodů) $D = 1$
5. (17 bodů) $L = 0$, $R = N-1$
6. (19 bodů) Hodnota D je stejná ve všech voláních funkce `max_towers`.
7. (23 bodů) Žádná další omezení.

Ukázkový grader

Ukázkový grader načítá vstup v následujícím formátu:

- řádek 1: $N\ Q$
- řádek 2: $H[0]\ H[1]\ \ldots\ H[N-1]$
- řádek $3+j$ ($0 \leq j \leq Q-1$): $L\ R\ D$ pro j-tou otázku.

Ukázkový grader vypíše vaše odpovědi v následujícím formátu:

- řádek $1+j$ ($0 \leq j \leq Q-1$): návratová hodnota j-tého volání funkce `max_towers`