

Фестивал

Найра е на фестивал, на който играе игра с прекрасна награда - пътешествие до Червената лагуна Колорада. В играта се използват монети за придобиване на купони. Придобиването на купон може да добави допълнителни монети. Целта е да се придобият възможно най-много купони.

Найра започва играта с A монети. Има общо N купона, номерирани с числата от 0 до $N - 1$. Найра трябва да плати $P[i]$ монети ($0 \leq i < N$) за придобиването на купон i (тя трябва да има поне $P[i]$ монети преди придобиването). Тя може да придобие всеки купон най-много веднъж.

Също така всеки купон i ($0 \leq i < N$) има **вид**, означен с $T[i]$, което е цяло число **между 1 и 4, включително**. След като Найра придобие купон i , останалият брой монети, с които разполага, се умножават по $T[i]$. По-точно, ако тя има X монети по някое време на играта и придобие купон i (за което е нужно $X \geq P[i]$), то тя ще има $(X - P[i]) \cdot T[i]$ монети след придобиването.

Вашата задача е да намерите кои купони трябва да придобие Найра, и в какъв ред, за да максимизира общия брой **купони**, с които разполага в края на играта. Ако има повече от една възможна поредица от придобивания, която постига това, можете да намерите която и да е от тях.

Детайли по имплементацията

Трябва да имплементирате следната функция.

```
std::vector<int> max_coupons(int A, std::vector<int> P,  
                             std::vector<int> T)
```

- A : първоначалният брой монети, с които разполага Найра.
- P : масив с големина N , задаващ цените на купоните.
- T : масив с големина N , задаващ видовете на купоните.
- Тази функция се извиква точно веднъж за всеки тест.

Тази функция трябва да върне масив R , описващ придобиванията на Найра по следния начин:

- Големината на R трябва да е максималният брой купони, които тя може да придобие.
- Елементите на масива трябва да са индексите на купоните, които тя ще придобие, в хронологичен ред. Това означава, че тя първо ще придобие купон $R[0]$, след това ще придобие купон $R[1]$ и т.н.
- Всички елементи на R трябва да са уникални.

Ако не могат да бъдат придобити купони, то R трябва да е празен масив.

Ограничения

- $1 \leq N \leq 200\,000$
- $1 \leq A \leq 10^9$
- $1 \leq P[i] \leq 10^9$ за всяко i , спазващо $0 \leq i < N$.
- $1 \leq T[i] \leq 4$ за всяко i , спазващо $0 \leq i < N$.

Подзадачи

Подзадача	Точки	Допълнителни ограничения
1	5	$T[i] = 1$ за всяко i , спазващо $0 \leq i < N$.
2	7	$N \leq 3000$; $T[i] \leq 2$ за всяко i , спазващо $0 \leq i < N$.
3	12	$T[i] \leq 2$ за всяко i , спазващо $0 \leq i < N$.
4	15	$N \leq 70$
5	27	Найра може да придобие всички N купона (в някакъв ред).
6	16	$(A - P[i]) \cdot T[i] < A$ за всяко i , спазващо $0 \leq i < N$.
7	18	Няма допълнително ограничения.

Примери

Пример 1

Нека разгледаме следното извикване.

```
max_coupons(13, [4, 500, 8, 14], [1, 3, 3, 4])
```

В началото Найра разполага с $A = 13$ монети. Тя може да придобие 3 купона в следния ред:

Купон	Цена на купона	Вид на купона	Брой монети след придобиването
2	8	3	$(13 - 8) \cdot 3 = 15$
3	14	4	$(15 - 14) \cdot 4 = 4$
0	4	1	$(4 - 4) \cdot 1 = 0$

В този пример не е възможно Найра да придобие повече от 3 купона и поредицата, описана по-горе, е единственият начин за придобиване на 3 от тях. Затова функцията връща $[2, 3, 0]$.

Пример 2

Нека разгледаме следното извикване.

```
max_coupons(9, [6, 5], [2, 3])
```

Найра може да придобие двата купона в какъвто и да е ред. Затова връща $[0, 1]$ или $[1, 0]$.

Пример 3

Нека разгледаме следното извикване.

```
max_coupons(1, [2, 5, 7], [4, 3, 1])
```

В този пример Найра разполага с една монета, което не е достатъчно за придобиването на никой от купоните. Затова функцията трябва да върне $[]$ (празен масив).

Локален грейдър

Формат на входа:

```
N A
P[0] T[0]
P[1] T[1]
...
P[N-1] T[N-1]
```

Формат на изхода:

```
S
R[0] R[1] ... R[S-1]
```

Тук S е големината на масива R , върнат от функцията `max_coupons`.