Souvenirs

Ամարուն հուշանվերներ է գնում արտասահմանյան խանութում: Կան N **տեսակի** հուշանվերներ: Խանութում յուրաքանչյուր տեսակի անսահման շատ հուշանվերներ կան:

Հուշանվերի յուրաքանչյուր տեսակ ունի ֆիքսված գին: Մասնավորապես, i տիպի հուշանվերը ($0 \leq i < N$) ունի $P[i]$ մետաղադրամ գին, որտեղ $P[i]$ դրական ամբողջ թիվ է:

Ամարուն գիտի, որ հուշանվերների տեսակները դասավորված են գնի նվազման կարգով, և որ հուշանվերների գները տարբեր են: Մասնավորապես, $P[0] > P[1] > \cdots > P[N-1] > 0$: Ավելին, նա կարողացավ իմանալ $P[0]$ ի արժեքը: Դժբախտաբար, Ամարուն հուշանվերների գների վերաբերյալ որևէ այլ տեղեկություն չունի:

Հուշանվերներ գնելու համար Ամարուն մի շարք գործարքներ կկատարի վաճառողի հետ:

Յուրաքանչյուր գործարք բաղկացած է հետևյալ քայլերից.

1. Ամարուն վաճառողին է հանձնում որոշակի (դրական) քանակությամբ մետաղադրամներ:
2. Վաճառողը այս մետաղադրամները դնում է կույտի մեջ՝ սեղանի վրա, հետևի սենյակում, որտեղ Ամարուն չի կարող տեսնել նրանց:
3. Վաճառողը յուրաքանչյուր հուշանվերի տեսակը դիտարկում է $0, 1, \ldots, N-1$ այդ հերթականությամբ՝ մեկ առ մեկ: Յուրաքանչյուր տեսակը դիտարկվում է **ճիշտ մեկ** գործարքի համար:
   - Երբ դիտարկվում է i տեսակի հուշանվերը, եթե կույտում մետաղադրամների ընթացիկ քանակը առնվազն $P[i]$ է, ապա
     - վաճառողը կույտից հանում է $P[i]$ մետաղադրամները, և
     - Վաճառողը սեղանին դնում է i տեսակի մեկ հուշանվեր:
4. Վաճառողը Ամարուին տալիս է կույտում մնացած բոլոր մետաղադրամները և սեղանին դրված բոլոր հուշանվերները:

Ուշադրություն դարձրեք, որ գործարքի սկսվելուց առաջ սեղանին մետաղադրամներ կամ հուշանվերներ չկան:

Ձեր խնդիրն է հանձնարարել Ամարուին կատարել որոշակի թվով գործարքներ այնպես, որ՝

- յուրաքանչյուր գործարքի ժամանակ նա գնի **առնվազն մեկ** հուշանվեր, և
- ընդհանուր առմամբ նա գնի **ճիշտ** ii տեսակի հուշանվերներ, յուրաքանչյուր i համար այնպիսին, որ $0 \leq i < N$: Նկատի ունեցեք, որ սա նշանակում է, որ Ամարուն չպետք է գնի 0 տեսակի որևէ հուշանվեր:

Ամարուն պարտավոր չէ նվազագույնի հասցնել գործարքների քանակը և ունի մետաղադրամների անսահմանափակ մատակարարում:

Իրականացման մանրամասներ

Դուք պետք է իրականացնեք հետևյալ ֆունկցիան.

```
void buy_souvenirs(int N, long long P0)
```

- N. հուշանվերների տեսակների քանակը:
- $P0$. $P[0]$ ի արժեքը:
- Այս ֆունկցիան կանչվում է ճիշտ մեկ անգամ յուրաքանչյուր թեստի համար:

Վերոնշյալ ֆունկցիան կարող է հետևյալ ֆունկցիայի կանչեր կատարել որպեսզի Ամարուին գործարք կատարելու հրահանգ տա՝

```
std::pair<std::vector<int>, long long> transaction(long long M)
```

- M. Ամարուի կողմից վաճառողին տրված մետաղադրամների քանակը:
- Ֆունկցիան վերադարձնում է զույգ: Զույգի առաջին տարրը L զանգվածն է, պարունակում է գնված հուշանվերների տեսակները (աճման կարգով): Երկրորդ տարրը ամբողջ թիվ է R , գործարքից հետո Ամարուին վերադարձված մետաղադրամների քանակը:
- Պահանջվում է, որ $P[0] > M \geq P[N-1]$: $P[0] > M$ պայմանը երաշխավորում է, որ Ամարուն չի գնի 0 տիպի որևէ հուշանվեր, իսկ $M \geq P[N-1]$ պայմանը երաշխավորում է, որ Ամարուն կգնի առնվազն մեկ հուշանվեր: Եթե այս պայմանները չեն բավարարվում, ձեր լուծումը կստանա `Output isn't correct: Invalid argument` վճիռը: Նկատի ունեցեք, որ ի տարբերություն $P[0]$-ի, $P[N-1]$ արժեքը մուտքային տվյալներում չի տրամադրվում:
- Յուրաքանչյուր թեստի համար ֆունկցիան կարող է կանչվել առավելագույնը 5000 անգամ:

Գրեյդերի վարքագիծը **հարմարվողական չէ**: Սա նշանակում է, որ գների P հաջորդականությունը ֆիքսված է նախքան `buy_souvenirs` կանչելը:

Սահմանափակումներ

- $2 \leq N \leq 100$
- $1 \leq P[i] \leq 10^{15}$ յուրաքանչյուր i համար, որտեղ $0 \leq i < N$ ։
- $P[i] > P[i+1]$ յուրաքանչյուր i համար, որտեղ $0 \leq i < N-1$ ։

Ենթախնդիրներ

Ենթախնդիր	Միավոր	Լրացուցիչ սահմանափակումներ
1	4	$N = 2$
2	3	$P[i] = N - i$ յուրաքանչյուր i համար, որտեղ $0 \leq i < N$.
3	14	$P[i] \leq P[i+1] + 2$ յուրաքանչյուր i համար, որտեղ $0 \leq i < N-1$ ։
4	18	$N = 3$
5	28	$P[i+1] + P[i+2] \leq P[i]$ յուրաքանչյուր i համար, որտեղ $0 \leq i < N-2$. $P[i] \leq 2 \cdot P[i+1]$ յուրաքանչյուր i համար, որտեղ $0 \leq i < N-1$.
6	33	Լրացուցիչ սահմանափակումներ չկան։

Օրինակ

Դիտարկենք հետևյալ կանչը։

```
buy_souvenirs(3, 4)
```

Կան $N = 3$ տեսակի հուշանվերներ և $P[0] = 4$ ։ Նկատի ունեցեք, որ P գների միայն երեք հնարավոր հաջորդականություն կա՝ $[4, 3, 2]$, $[4, 3, 1]$, և $[4, 2, 1]$ ։

Ենթադրենք, որ `buy_souvenirs` կանչում է `transaction(2)` ։ Ենթադրենք, որ կանչը վերադարձնում է $([2], 1)$, ինչը նշանակում է, որ Ամարուն գնել է մեկ հուշանվեր՝ 2 տեսակի և վաճառողը նրան վերադարձրեց 1 մետաղադրամը։ Նկատենք, որ սա թույլ է տալիս մեզ եզրակացնել, որ $P = [4, 3, 1]$, քանի որ՝

- $P = [4, 3, 2]$ դեպքում, `transaction(2)`-ը կվերադարձներ $([2], 0)$ ։
- $P = [4, 2, 1]$ դեպքում, `transaction(2)`-ը կվերադարձներ $([1], 0)$ ։

Այնուհետև `buy_souvenirs` կարող է կանչել `transaction(3)` , որը վերադարձնում է $([1], 0)$, այսինքն՝ Ամարուն գնել է մեկ հուշանվեր՝ 1 տեսակի և վաճառողը նրան վերադարձրեց

0 մետաղադրամ։ Մինչ օրս, ընդհանուր առմամբ, Նա գնել է մեկ հուշանվեր՝ 1 տեսակի և մեկ հուշանվեր՝ 2 տեսակի։

Վերջապես, `buy_souvenirs` կարող է կանչել `transaction(1)` , որը վերադարձնում է $([2], 0)$, ինչը նշանակում է, որ Ամարուն գնել է մեկ հուշանվեր՝ 2 տեսակի։ Նկատի ունեցեք, որ այստեղ կարող էինք օգտագործել նաև `transaction(2)` ը։ Այս պահին, ընդհանուր առմամբ Ամարուն ունի մեկ 1 տեսակի հուշանվեր և երկու 2 տեսակի հուշանվեր՝ ինչպես պահանջվում էր։

Նմուշային գրեյդեր

Մուտքային տվյալների ձևաչափը՝

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

Ելքային տվյալների ձևաչափը՝

```
Q[0] Q[1] ... Q[N-1]
```

Այստեղ $Q[i]-ն i$ տեսակի հուշանվերների ընդհանուր քանակն է, որը գնվել է։ յուրաքանչյուր i համար այնպիսին, որ $0 \leq i < N$ ։