

Partition

Թեմուրը և Նրա օգնական Ուլուգբեկը կախարդական հնարք են պատրաստում *Ուլգբեկստանը տաղանդ ունի* շոուի համար: Հնարքը պտտվում է Թեմուրի կողմից հետևյալ **տրոհման խնդրի** լուծման շուրջ. տրված դրական ամբողջ թվերի հավաքածուն բաժանել K ոչ դատարկ խմբերի այնպես, որ յուրաքանչյուր խմբի տարրերի գումարը նույնը լինի: Այլ կերպ ասած, հավաքածուի յուրաքանչյուր ամբողջ թիվ պետք է պատկանի K խմբերից ճիշտ մեկին, և յուրաքանչյուր խմբի ներսում եղած թվերի գումարները պետք է հավասար լինեն: Օրինակ, եթե տրված հավաքածուն $[2, 1, 6, 4, 5]$ է և $K = 3$, վավեր բաժինը կլինի $[2, 4]$, $[1, 5]$ և $[6]$: Այս դեպքում յուրաքանչյուր խմբի տարրերի գումարը ճիշտ 6 է:

Հնարքն իրականացվում է հետևյալ կերպ.

- Ուլուգբեկը և Թեմուրը մտնում են առանձին խցիկներ և չեն կարողանում շփվել միմյանց հետ:
- Ժյուրիի անդամը Ուլուգբեկին տալիս է N ամբողջ թվերից բաղկացած A զանգված՝ $A[0], A[1], \dots, A[N - 1]$, որտեղ յուրաքանչյուր արժեք 1-ից մինչև M միջակայքից է, և K -ն:
- Ուլուգբեկը ընտրում է առավելագույնը $K - 1$ (պարտադիր չէ, տարբեր) հատ նոր թիվ A զանգվածին ավելացնելու համար: Յուրաքանչյուր ավելացված ամբողջ թիվ նույնպես պետք է լինի 1-ից մինչև M ներառյալ ամբողջ թիվ:
- Ժյուրին Ուլուգբեկի կողմից ընտրված ամբողջ թվերը կցում է սկզբնական զանգվածին: Այս ընդլայնված զանգվածը այնուհետև **դասավորվում է չնվազման կարգով** և տրվում է Թեմուրին K արժեքի հետ միասին:
- Այնուհետև Թեմուրը պետք է լուծի այս ընդլայնված, տեսակավորված զանգվածի համար նշված տրոհման խնդիրը:

Ձեր խնդիրն է մշակել և իրականացնել ռազմավարություն Թեմուրի և Ուլուգբեկի համար: Կարելի է ապացուցել, որ տրված սահմանափակումների դեպքում գոյություն ունի ռազմավարություն, որը թույլ է տալիս նրանց հաջողությամբ լուծել տրոհման խնդիրը անկախ ժյուրիի տրամադրած A զանգվածից:

Իրականացման մանրամասներ

Դուք պետք է իրականացնեք երկու ֆունկցիա .

Ուլուգբեկի համար դուք պետք է իրականացնեք հետևյալ ֆունկցիան՝

```
std::vector<int> add_numbers(std::vector<int> A, int K, int M)
```

- A . N երկարությամբ զանգված, որը ներկայացնում է Ուլուգբեկին տրված սկզբնական զանգվածը:
- K . խմբերի նպատակային քանակը, նկատել, որ Ուլուգբեկը կարող է ավելացնել զանգվածին մինչև $K - 1$ հատ թիվ:
- M . յուրաքանչյուր սկզբնական և նոր ավելացված ամբողջ թվի համար թույլատրելի առավելագույն արժեքը:
- Այս ֆունկցիան կանչվում է ճիշտ մեկ անգամ յուրաքանչյուր թեստի համար:

Ֆունկցիան պետք է վերադարձնի C զանգված, որը պարունակում է այն ամբողջ թվերը, որոնք Ուլուգբեկը ցանկանում է ավելացնել սկզբնական զանգվածին: Ենթադրենք S -ը C զանգվածի երկարությունն է:

- S պետք է լինի առավելագույնը $K - 1$:
- C -ի յուրաքանչյուր տարր պետք է լինի 1 -ից մինչև M (ներառյալ) միջակայքում:

Թեմուդի համար դուք պետք է իրականացնեք հետևյալ ֆունկցիան՝

```
std::vector<int> find_partition(std::vector<int> B, int K)
```

- B . $N + S$ երկարությամբ զանգված, որը պարունակում է ինչպես A -ից ստացված սկզբնական ամբողջ թվերը, այնպես էլ Ուլուգբեկի կողմից ավելացված ամբողջ թվերը: B զանգվածի տարրերը դասավորված են չնվազման կարգով:
- K . խմբերի նպատակային քանակը:
- Այս ֆունկցիան կանչվում է ճիշտ մեկ անգամ յուրաքանչյուր թեստի համար:

Ֆունկցիան պետք է վերադարձնի P զանգված, որը ներկայացնում է B զանգվածի բաժանումը K խմբերի միջև:

- P -ի երկարությունը պետք է լինի $N + S$:
- Յուրաքանչյուր j -ի համար, որտեղ $0 \leq j < N + S$, $P[j]$ -ն նշանակում է այն խումբը, որին պատկանում է $B[j]$ -ն:
- Խմբերը պետք է համարակալվեն 0 -ից մինչև $K - 1$, ինչը նշանակում է, որ $0 \leq P[j] < K$ յուրաքանչյուր $0 \leq j < N + S$ -ի համար:
- 0 -ից մինչև $K - 1$ ներառյալ յուրաքանչյուր i համար պետք է լինի առնվազն մեկ j տարր ($0 \leq j < N + S$) այնպիսին, որ $P[j] = i$:
- K խմբերից յուրաքանչյուրին վերագրված տարրերի գումարը պետք է նույնական լինի:

Իրական թեստավորման ժամանակ վերը նշված ֆունկցիաները կանչող ծրագիրը կատարվում է **ճիշտ երկու անգամ**:

- Ծրագրի առաջին գործարկման ժամանակ
 - `add_numbers` ֆունկցիան կանչվում է ճիշտ մեկ անգամ:

- Վերադարձված զանգվածը մշակվում է թեստավորման համակարգի կողմից՝ B զանգվածը ստանալու համար:
- Ծրագրի երկրորդ գործարկման ժամանակ
 - `find_partition` ֆունկցիան կանչվում է ճիշտ մեկ անգամ:

Սահմանափակումներ

- $3 \leq N \leq 100\,000$
- $2 \leq K \leq 100\,000$
- $K \leq N$
- $1 \leq M \leq 10^9$
- $1 \leq A[i] \leq M$ յուրաքանչյուր i համար, որտեղ $0 \leq i < N$.

Ենթախնդիրներ

Ենթախնդիր	Միավոր	Լրացուցիչ սահմանափակումներ
1	5	$N = 3$
2	4	$M = 1$
3	7	$M \leq 2$
4	10	$N \leq 10$
5	7	$K = 2$
6	12	$K \leq 3$
7	17	$K \leq 10$
8	20	$K \leq 100$
9	18	Լրացուցիչ սահմանափակումներ չկան

Օրինակ

Դիտարկենք հետևյալ կանչը.

```
add_numbers([8, 2, 9, 6, 1, 5, 5], 3, 9)
```

Այս օրինակում $A = [8, 2, 9, 6, 1, 5, 5]$: Մենք ուզում ենք բաժանել թվերը $K = 3$ խմբերի: Ուղղաբեկը կարող է ավելացնել թվեր 1-ից մինչև $M = 9$ միջակայքից:

Ֆունկցիան կարող է վերադարձնել $[5, 4]$, ինչը նշանակում է, որ Ուղղաբեկը որոշում է ավելացնել $S = 2$ նոր ամբողջ թվեր՝ 5 և 4: Դրանք համարվում են վավեր, քանի որ երկու տարրերն էլ գտնվում են 1-ի և $M = 9$ -ի միջև:

Այնուհետև ընդլայնված զանգվածը դասավորվում է չնվազման կարգով և փոխանցվում է Թեմուրին հետևյալ կանչով՝

```
find_partition([1, 2, 4, 5, 5, 5, 6, 8, 9], 3)
```

Մենք կարող ենք այս զանգվածի ամբողջ թվերը բաժանել երեք խմբի՝ $[1, 5, 9]$, $[2, 5, 8]$ և $[4, 5, 6]$: Նկատենք, որ յուրաքանչյուր խմբի գումարը հավասար է 15-ին: Այս բաժանումը հաղորդելու համար ֆունկցիան պետք է վերադարձնի $[0, 1, 2, 0, 1, 2, 2, 1, 0]$ զանգվածը:

Գրեյդերի նմուշ

Մուտքային տվյալներ

```
N K M
A[0] A[1] ... A[N-1]
```

Ելքային տվյալներ

`add_numbers` ֆունկցիայի կանչի ավարտից հետո գրեյդերի նմուշը

- Հաշվարկում է սորտավորված B զանգվածը:
- Տպում է C և B զանգվածները հետևյալ ձևաչափով, որին հաջորդում է դատարկ տող:

```
S
C[0] C[1] ... C[S-1]
B[0] B[1] ... B[N+S-1]
```

Երբ ավարտվում է `find_partition` ֆունկցիայի կանչը, գրեյդերը տպում է՝

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

Այստեղ L -ը `find_partition`-ի կոդմիջ վերադարձված P զանգվածի երկարությունն է: