

Task 1. Heavy coins

Հարթավայրային թագավորությունում կար մի հոյակապ պարիսպ, որը դարեր շարունակ պաշտպանում էր նրա տարածքը արշավանքներից: Սակայն ուղիղ մեկ տարի առաջ մի ավազակախումբ ոչնչացրել է այն: Այժմ արքայադուստր Մարիան ծրագրել է վարձել հմուտ շինարարներ՝ պատը իր սկզբնական շուքով վերականգնելու համար: Շինարարներին վարձելու համար նա գնաց թագավորական գանձատուն և այնտեղ գտավ $N = 2^K$ մետաղադրամ, որոնցից յուրաքանչյուրն ունի 0-ից $N - 1$ չկրկնվող համար: Պալատի հաշվապահը նրան տեղեկացրեց, որ կան երկու տեսակի մետաղադրամներ՝ թեթև և ծանր, ընդ որում բոլոր ծանր մետաղադրամները հավասարապես ծանր են, ինչպես նաև բոլոր թեթև մետաղադրամները: Նաև, ըստ հաշվապահի, գանձարանում կա առնվազն մեկ ծանր և թեթև մետաղադրամ:

Նոր պարիսպը կառուցողների հետ համաձայնեցված պայմանների համաձայն՝ արքայադուստր Մարիան պետք է նրանց վճարի միայն ծանր մետաղադրամներով, և մինչ աշխատանքը սկսելը, նա պետք է որպես կանխավճար մեկ ծանր մետաղադրամ վճարի: Խնդիրն այն է, որ հաշվապահը չգիտի, թե մետաղադրամներից քանիսն են և որոնք են ծանր: Բարեբախտաբար, նա ունի կշեռքներ, որոնցով կարող է համեմատել նույն թվով մետաղադրամներից բաղկացած երկու հավաքածուների համակցված քաշը: Այժմ հաշվապահը պետք է համոզվի, որ թագավորական գանձարանում ծանր մետաղադրամների քանակը բավարար կլինի վճարման համար և պետք է գտնի այդպիսի մետաղադրամ, որով հնարավոր կլինի վճարել կանխավճարը: Այնուամենայնիվ, դա նրանից չափազանց շատ ժամանակ կխլի, իսկ արքայադուստր Մարիան ցանկանում է, որ նոր պատի շինարարությունը սկսվի որքան հնարավոր է շուտ, ուստի նա դիմել է ձեզ օգնության համար:

Խնդիր

Գրեք `coins` ծրագիր, իրականացնելով `count_heavy` ֆունկցիան, որը կկոմպիլացվի ժյուրիի ծրագրի հետ, այն ժյուրիի ծրագրի հետ հաղորդակցվելով պետք է գտնի թագավորական գանձատանը ծանր մետաղադրամների ընդհանուր քանակը, ինչպես նաև այդպիսի մեկ մետաղադրամի համարը:

Իրականացման մանրամասներ

Ձեր ֆունկցիան պետք է ունենա այսպիսի ձևաչափ.

`pair<int, int> count_heavy(int k)`

Այն ստանում է որպես պարամետր K թիվը (հիշեցնենք, որ մետաղադրամների քանակը 2^K է) և պետք է վերադարձնի երկու ամբողջ թվերի զույգ՝ ծանր մետաղադրամների քանակը և դրանցից մեկի համարը:

Ժյուրիի ծրագրի հետ հաղորդակցվելու համար ձեզ տրամադրվում է հետևյալ ֆունկցիան.

`int weigh(const vector<int> &a, const vector<int> &b)`

Այն ունի երկու պարամետր, վեկտորները պետք է պարունակեն այն մետաղադրամների համարները, որոնք պետք է կշռել: Երկու վեկտորները պետք է պարունակեն հավասար քանակով թվեր, կրկնվող թվեր չպետք է լինեն: Ֆունկցիան կվերադարձնի.

- ☐ 0, եթե մետաղադրամների երկու բազմությունների քաշերը հավասար են;
- ☐ -1 , եթե a վեկտորում պահվող համարներով մետաղադրամների ընդհանուր քաշն ավելի փոքր է, քան b վեկտորում պահվողներինը;
- ☐ $+1$, եթե a վեկտորում պահվող համարներով մետաղադրամների ընդհանուր քաշն ավելի մեծ է, քան b վեկտորում պահվողներինը:

Նկատենք, որ `weigh` ֆունկցիայի բարդությունը համեմատական է վեկտորների տարրերի ընդհանուր քանակին:

Դուք պետք է ստուգող համակարգին ուղարկեք `coins.cpp` ֆայլը, որը պարունակում է `count_heavy` ֆունկցիան: Այն կարող է, ձեր ծրագրի աշխատանքի համար անհրաժեշտ այլ կոդ և

Ֆունկցիաներ պարունակել, բայց չպետք է պարունակի *main* ֆունկցիա: Նաև, դուք չպետք է կարդաք ստանդարտ մուտքից և տպեք ստանդարտ ելքում: Նաև ձեր ծրագիրը պետք է include անի `coins.h` header ֆայլը.

```
#include "coins.h"
```

Սահմանափակումներ

$$1 \leq K \leq 20$$

Գնահատումը

Եթե, ձեր ծրագիրը տվյալ թեստի համար տա ճիշտ պատասխան անելով `weigh` ֆունկցիայի ոչ ավելի քան 10 000 կանչ, այդ թեստի համար կհաշվարկվի հետևյալ գործակիցը.

$$coef = \min\left(1, \frac{11}{9} * \left(\frac{author+1}{yours+1}\right)\right),$$
 որտեղ *author*-ը հեղինակների լուծումներից մեկի՝ *weigh*

ֆունկցիայի կանչերի քանական է, իսկ *yours*-ը ձեր ծրագրի կատարած `weigh` ֆունկցիայի կանչերի քանակն է: Հակառակ դեպքում այդ թեստի համար *coef*-ի արժեքը կլինի 0: **Երաշխավորվում է, որ բոլոր թեստերում *author*-ի արժեքը փոքր է 1 000-ից:**

Յուրաքանչյուր ենթախնդիր բաղկացած է որոշակի քանակով խմբերից, յուրաքանչյուր խմբում կա ճիշտ երեք հատ թեստ: Յուրաքանչյուր խումբ ստուգվում է մյուսներից **անկախ**, և դրա համար ձեր ստացած միավորը հավասար է այդ խմբի համար նախատեսված միավորը բազմապատկած այդ խմբի թեստերի *coef* գործակիցներից փոքրագույնով:

Ենթախնդիրներ

Ենթախնդիր	Միավոր	Այլ սահմանափակումներ
1	5	$K \leq 5$
2	10	Կա միայն մեկ ծանր և մեկ թեթև մետաղադրամ:
3	20	Կան ոչ ավել, քան K ծանր և ոչ ավել, քան K թեթև մետաղադրամներ:
4	25	$K \leq 10$
5	40	Լրացուցիչ սահմանափակումներ չկան

Հաղորդակցման նմուշ

Մրցույթի մասնակցի ֆունկցիան	Ժյուրիի ծրագիրը
	Կանչում է <code>count_heavy(2)</code>
Կանչում է <code>weigh({0, 1}, {2, 3})</code>	Վերադարձնում է -1
Կանչում է <code>weigh({1, 2}, {0, 3})</code>	Վերադարձնում է +1
Կանչում է <code>weigh({1}, {2})</code>	Վերադարձնում է 0
Վերադարձնում է {3, 2}	

Լոկալ թեստավորում

Լոկալ թեստավորման համար ձեզ տրամադրվում են `coins.h` և `Lgrader.cpp` ֆայլերը: Ձեր `coins.cpp` ֆայլը և տրամադրված երկու ֆայլերը տեղադրեք միևնույն ֆոլդերում: Ապա կոմպիլացրեք երեք ֆայլերն իրար հետ: Այդ կերպ դուք կստանաք ծրագիր ձեր ֆունկցիայի ճշտությունը ստուգելու համար: Այդ ծրագիրը պահանջելու է տվյալների հետևյալ հաջորդականությունը.

- Առաջին տողում. մեկ դրական ամբողջ թիվ՝ K -ի արժեքը:

- Երկրորդ տողում. 2^K թվեր՝ 1-եր և 0-ներ, որոնք ցույց են տալիս համապատասխան մետաղադրամը ծանր է, թե թեթև:

Ծրագիրը կարտածի տեղեկատվություն *weigh* ֆունկցիայի կանչերի վերաբերյալ, ձեր *count_heavy* ֆունկցիայի վերադարձրած պատասխանը և հաղորդագրություն, որը նշում է, այդ պատասխանը ճիշտ է, թե սխալ: