

Задатак Jimbo

Улазна датотека `stdin`
Изразна датотека `stdout`



(a) Прошлост



(b) Садашњост



(c) Будућност

Значење тарот карата задатка *Jimbo*

Гими је открио нову игру звану *Talatro*. У овој игри Гими има шпил од N карата и његов циљ је да максимизује резултат који добије одигравањем тачно 5 карата.

На почету игре, резултат је одређен са две вредности:

- Бројем чипова, означеним са C , који је на почетку игре 0.
- Коефицијентом множења, означеним са M , који је на почетку игре 1.

Свака карта коју Гими одигра може да промени вредности C и M . Постоје 4 типа карата:

- 1 x : Увећава C за x . Другим речима, $C \leftarrow C + x$.
- 2 $x \ y$: Увећава C за x и увећава M за y . Другим речима, $C \leftarrow C + x, M \leftarrow M + y$.
- 3 $x \ z$: Увећава C за x и множи M са z . Другим речима, $C \leftarrow C + x, M \leftarrow M \cdot z$.
- 4 $x \ y \ z$: Увећава C за x , након чега увећава M за y , и на крају множи M са z . Другим речима, $C \leftarrow C + x, M \leftarrow (M + y) \cdot z$.

Крајњи резултат је једнак производу бројева C и M након што је Гими одиграо свих 5 карата. Гими бира редослед којим игра карте. Који је највећи могући резултат који Гими може да постигне?

Задатак

Написати програм који за дати број N (број карата у шпилу), као и за дати опис N карата налази највећи резултат који Гими може да постигне.

Улаз

Прва линија улаза садржи природни број N , број карата у шпилу. Свака од следећих N линија даје опис једне карте, где је прва вредност у линији t_i , која означава тип карте i . У зависности од типа карта, остале вредности у тој линији су:

- За $t_i = 1$, дата је вредност x_i .
- За $t_i = 2$, дате су вредности x_i и y_i .
- За $t_i = 3$, дате су вредности x_i и z_i .
- За $t_i = 4$, дате су вредности x_i, y_i и z_i .

Излаз

Прва и једина линија излаза садржи један природан број који представља највећи резултат који је Гими могао остварити.

Ограничења

- $5 \leq N \leq 1\,000$.
- $1 \leq x_i, y_i \leq 1\,000$, за свако $1 \leq i \leq N$.
- $2 \leq z_i \leq 100$, за свако $1 \leq i \leq N$.

#	Поени	Ограничења
1	8	$5 \leq N \leq 10$
2	10	$5 \leq N \leq 15$
3	14	$x_1 = x_2 = \dots = x_N$
4	18	Све карте типа 2 и 4 имају исту вредност y , док све карте типа 3 и 4 имају исту вредност z .
5	23	Не постоје карте типа $t = 4$.
6	27	Нема додатних ограничења.

Примери

stdin	stdout	Објашњења
6 1 3 1 5 2 1 1 3 1 2 4 1 1 2 3 1 3	324	Оптимални редослед играња карата је да се играју карте са индексима 2, 3, 5, 4 и 6 редом. Почетно: $C = 0, M = 1$. Након одигравања карте 2: $C = 5, M = 1$ Након одигравања карте 3: $C = 6, M = 2$ Након одигравања карте 5: $C = 7, M = 6$ Након одигравања карте 4: $C = 8, M = 12$ Након одигравања карте 6: $C = 9, M = 36$ Резултат на крају: $C \cdot M = 9 \cdot 36 = 324$. Другачији редослед играња карата може да доведе до мањег резултата.

Задатак Negotiable

Улазна датотека stdin
Излазна датотека stdout



(a) Прошлост



(b) Садашњост



(c) Будућност

Значење тарот карата задатка *Negotiable*

Задатак

Функција МЕХ неког низа бројева је најмањи ненегативни цео број који се не налази у том низу. На пример, $\text{MEH}(0, 4, 4, 1, 2) = 3$ и $\text{MEH}(1, 2, 3) = 0$.

Низ b чији су чланови $b_1, b_2, \dots, b_K$ називамо x -преговорљив ако важи $\text{MEH}(b_1, b_2, \dots, b_K) \leq x$.

За дати низ b дужине K , уочимо неки низ индекса $1 \leq i_1 < i_2 < \dots < i_s < K$. Овај низ одређује поделу низа b на $s + 1$ узастопних поднизова: $(b_1, b_2, \dots, b_{i_1})$, $(b_{i_1+1}, b_{i_1+2}, \dots, b_{i_2})$, $\dots$, $(b_{i_{s-1}+1}, b_{i_{s-1}+2}, \dots, b_{i_s})$, $(b_{i_s+1}, b_{i_s+2}, \dots, b_K)$. Ову поделу називамо x -попустљива партиција ако је сваки од $s + 1$ низова x -преговорљив. Дефинишемо величину партиције као број поднизова, односно $s + 1$.

Дат је низ a од N ненегативних целих бројева. Одредити за свако x од 1 до N , минималну величину x -попустљиве партиције низа a .

Улазни подаци

Прва линија стандардног улаза садржи један природан број N . Друга линија стандардног улаза садржи N ненегативних целих бројева - елементе низа A .

Излаз

Исписати N ненегативних целих бројева размаком раздвојених, који представљају одговор за свако x .

Ограничења

- $1 \leq N \leq 1\,000\,000$.
- $0 \leq a_i \leq N$.

#	Поени	Ограничења
1	13	$1 \leq N \leq 10$
2	19	Постоји највише 50 локалних минимума или максимума.
3	23	$0 \leq a_i \leq 10$
4	19	$1 \leq N \leq 5000$
5	12	$1 \leq N \leq 100\,000$
6	14	Без додатних ограничења.

Локални минимуми дефинисани су као вредности a_i за које важи $2 \leq i \leq N - 1$ и $a_{i-1} \geq a_i \leq a_{i+1}$. Локални максимуми дефинисани су као вредности a_i за које важи $2 \leq i \leq N - 1$ и $a_{i-1} \leq a_i \geq a_{i+1}$.

Примери

stdin	stdout	Објашњења
5 0 1 2 3 0	3 2 2 1 1	За $x = 1$, оптимална партиција је $[1, 1], [2, 3], [4, 5]$. За $x = 2$, оптимална партиција је $[1, 2], [3, 5]$. За $x = 3$, оптимална партиција је $[1, 3], [4, 5]$. За $x = 4$, оптимална партиција је $[1, 5]$. За $x = 5$, оптимална партиција је $[1, 5]$.

Задатак Permuturns

Улазна датотека stdin
Излазна датотека stdout



(a) Прошлост



(b) Садашњост



(c) Будућност

Значење тарот карата задатка *Permuturns*

Алин и Блин играју игру под називом Palatro. Они деле шпил од N карата нумерисаних од 1 до N . Алин на карту са бројем i пише број A_i , а Блин на карту са бројем i пише број B_i . Сваки играч мора написати сваки број од 1 до N тачно једном (дакле, низови A и B су пермутације).

Клин је судија у игри. Он одређује редослед у којем ће два играча откривати своје карте, дајући N захтева облика:

„Покажи карту са индексом C_1 ”, затим „Покажи карту са индексом C_2 ”, и тако даље. Овде $(C_1, C_2, \dots, C_N)$ представља пермутацију бројева од 1 до N .

На почетку, Клин сматра прву откривену карту (C_1) победничком картом. Сваки пут када се открије нова карта C_i , Клин проверава бројеве написане на њој: ако су и Алинов број и Блинов број на тој карти већи од бројева на тренутној победничкој карти, онда карта C_i постаје нова победничка карта.

Формалније, Клин користи следећи алгоритам да одреди победничку карту:

```
winner = C[1]
for i=2...N:
    if A[C[i]] > A[winner] and B[C[i]] > B[winner]:
        winner = C[i]
```

Задатак

За свако i од 1 до N , одредити на колико различитих начина Клин може дати захтеве тако да победничка карта буде карта i . Пошто одговор може бити веома велики, треба исписати његов остатак при дељењу са $10^9 + 7$.

Улаз

Први ред садржи цео број N . Други ред садржи N целих бројева — елементе пермутације A . Трећи ред садржи N целих бројева — елементе пермутације B .

Излаз

Исписати N целих бројева раздвојених размаком — одговор за свако i .

Ограничења

- $1 \leq N \leq 200\,000$.

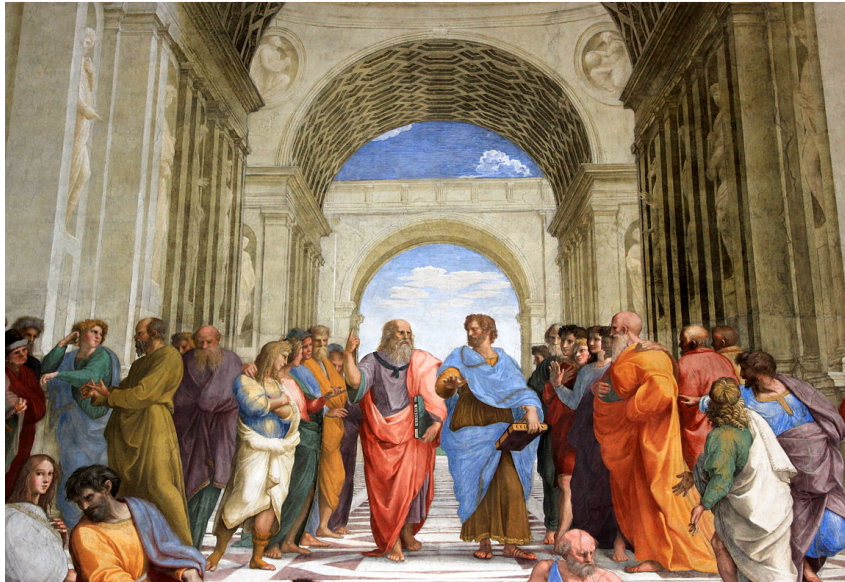
#	Посни	Ограничења
1	6	$A = 1, 2, \dots, N$ и $B = N - 1, N - 2, \dots, 1, N$
2	6	$1 \leq N \leq 10$
3	14	$1 \leq N \leq 20$
4	13	$1 \leq N \leq 500$
5	16	Низови A и B се генеришу тако да је свака пермутација подједнако вероватна.
6	24	$1 \leq N \leq 5000$
7	21	Без додатних ограничења

Примери

stdin	stdout	Објашњења
3 2 1 3 3 2 1	4 0 2	$C = [1, 2, 3] \Rightarrow$ На крају алгорита, $winner = 1$ $C = [1, 3, 2] \Rightarrow$ На крају алгорита, $winner = 1$ $C = [2, 1, 3] \Rightarrow$ На крају алгорита, $winner = 1$ $C = [2, 3, 1] \Rightarrow$ На крају алгорита, $winner = 1$ $C = [3, 1, 2] \Rightarrow$ На крају алгорита, $winner = 3$ $C = [3, 2, 1] \Rightarrow$ На крају алгорита, $winner = 3$

Задатак Atena

Улазна датотека `stdin`
Излазна датотека `stdout`



Платон и Аристотел, добри пријатељи, изгубили су се у Атини. Могло би се рећи да имају платонско пријатељство. Ипак, удаљили су се толико да нису могли директно да се чују чак ни када би викали. Зато им је био потребан посредник – Питагора. Док двојица пријатеља мирују, Питагора може својим знањем да израчуна површину зоне у којој мора да се налази да би чуо обоје, и да им пренесе информацију Морзевом азбуком у IEEE 754 формату. Пошто је Платон стар, није могао да се креће брзо као његов ученик Аристотел, па је остао непомичан.

У овом задатку ви сте у положају Аристотела. Комисија има скривени круг $\mathcal{C}_1$ са центром (X, Y) (који представља Платонову позицију) и полупречником R (на којем се може чути филозофов глас). Такмичар треба да пронађе бројеве (X, Y, R) користећи што мањи број питања следећег типа:

Нека је $\mathcal{C}_2$ круг са центром (x', y') и полупречником r' (који представља Аристотелов положај и капацитет плућа). Која је површина пресека кругова $\mathcal{C}_1$ и $\mathcal{C}_2$?

Бројеве x', y', r' бира такмичар, уз следећа ограничења:

- $-10^7 \leq x' \leq 10^7$
- $-10^7 \leq y' \leq 10^7$
- $0 < r' \leq 10^7$

Под површином пресека два круга подразумевамо површину заједничког дела ког одређују ти кругови.

Задатак

Написати програм који ће што прецизније и са што мање питања одредити координате круга. Поени ће се додељивати на основу броја постављених питања и тачности добијеног решења.

Напомена: Такмичарима ће бити обезбеђено неколико помоћних геометријских функција ради поједностављења имплементације.

Детаљи имплементације

Такмичар треба да имплементира само функцију:

```
std::vector<double> solve();
```

Функција враћа вектор $v = (v_1, v_2, v_3)$ од 3 елемента, тако да важи $X = v_1$, $Y = v_2$, $R = v_3$. Такмичар не треба да имплементира `main()` функцију.

Да би поставио питање, такмичар може позвати функцију:

```
double query(double x, double y, double r);
```

Функција враћа површину пресека дела одређеног кругом са центром (x, y) и полупречником r и скривеног круга $\mathcal{C}_1$ који поседује комисија.

Приложени фајлови

Такмичари имају приступ следећим фајловима:

- `atena.h` – хедер који дефинише интерактивне функције (`solve` и `query`) потребне за имплементацију.
- `atena.cpp` – пример имплементације који извршава један упит и увек враћа круг $(1, 2, 3)$.
- `lgrader.cpp` – грејдер за тестирање који омогућава локално проверавање решења. За званично оцењивање користиће се други грејдер.
- `geometry.h` — библиотека са више геометријских функција ради лакше имплементације.

Коришћење lgrader-a у Code::Blocks-у

Да бисте користили `lgrader.cpp` у Code::Blocks-у:

1. Креирајте нови пројекат.
2. Додајте фајл под именом `solutie.cpp` (изаберите `File > New > Empty File`), у којем ћете куцати своје решење.
3. Додајте празан фајл под именом `atena.h` и копирајте у њега садржај приложеног фајла `atena.h`.
4. Копирајте садржај из `lgrader.cpp` у `main.cpp` како бисте симулирали покретање решења.

Након тога, у `solutie.cpp` напишите имплементацију функције `solve`. За тестирање, можете најпре копирати код из `atena.cpp` у `solutie.cpp`. Притисните F9 или изаберите **Build and Run** за компајларање и покретање програма. При извршавању, програм чита вредности X, Y и R , симулира интеракцију између `lgrader`-а и решења такмичара, приказује резултате упита преко `query` функције и проверава тачност одговора из `solve`.

Ограничења

- Нека је $\mathcal{C}_1$ скривени круг комисије са центром (X, Y) и полупречником R .
- $-10^6 \leq X \leq 10^6$
- $-10^6 \leq Y \leq 10^6$
- **Пажња!** $10^4 \leq R \leq 10^6$

Бодовање

Нека је $D = (x - X)^2 + (y - Y)^2 + (r - R)^2$ грешка решења, а Q број постављених питања.

Коефицијенти C_D и C_Q дефинисани су као:

$$C_D = \begin{cases} 1 & D < 10^{-5} \\ \frac{2 - \log_{10}(D)}{7} & 10^{-5} \leq D \leq 10^2 \\ 0 & D > 10^2 \end{cases}$$
$$C_Q = \begin{cases} 1 & Q \leq 20 \\ \left(\frac{70 - Q}{50}\right)^{1.3} \cdot 0.3 + 0.7 & 20 < Q \leq 70 \\ \frac{780 - 3Q}{1900} + 0.4 & 70 < Q \leq 260 \\ \frac{5000 - Q}{14960} + 0.15 & 260 < Q \leq 5000 \\ 0 & Q > 5000 \end{cases}$$

Ваш резултат за тест биће $100 \cdot C_D \cdot C_Q\%$ од максималног броја поена.

Гарантује се да ће за тестове вредне 20 поена центар скривеног круга лежати на 0x оси.

Пример интеракције

Скривени круг је $\mathcal{C}_1 (X = 17.00000, Y = 2.62331, R = 6.7890)$. Овај пример се неће појавити у тестовима јер R не задовољава доњу границу 10^4 .

Пример интеракције:

```
query(1, 1.5, 2.5)
```

Површина пресека између скривеног круга $\mathcal{C}_1$ и круга $(1, 1.5, 2.5)$ је 0.

```
query(15, 14, 7)
```

Површина пресека између круга $\mathcal{C}_1$ и круга $(15, 14, 7)$ је 11.429000978.

```
return {17.000001, 2.62331, 6.7891};
```

Иако враћене вредности нису у децималу идентичне правим, решење се сматра тачним јер је грешка мања од 10^{-5} .

Задатак Caterpillar 3

Улазна датотека `stdin`
Излазна датотека `stdout`

— Извини, на штампарској грешци.
Нисмо имали намеру да ти повредимо осећања.
— Често се нађемо у ситуацији у којој покушавамо да оправдамо све непоштене поступке у које се упуштамо само да бисмо могли да наставимо даље. Али, важно је у том процесу да не одлутамо од једине истине коју имамо — садашњег тренутка.
У настојању да поједноставимо историјске чињенице ради могуће јасније слике, можемо се наћи у ситуацији да уништавамо везе и заједнице. Шта би могло бити светије од садашњег тренутка, питамо те? У ком тренутку само уздржавање постаје проблематичан ревизионизам — и зашто не бисмо веровали да си ту границу већ прешао?

– Аутор заједно са **веома мађарском** (и можда гладном) гусеницом

Задатак

Размотримо битовски AND оператор (означаваћемо га са $\&$)¹.

Назовимо низ b бројева $b_1, b_2, \dots, b_k$ *самоописујућ* ако постоји индекс p ($1 \leq p \leq k$) такав да $b_1 \& b_2 \& \dots \& b_k = b_p$.

Дат је низ v који садржи N ненегативних целих бројева $v_1, v_2, \dots, v_N$. Одредити колико има парова индекса (l, r) који ограничавају *самоописујућ* подниз. Другим речима, израчунати број парова бројева (l, r) ($1 \leq l \leq r \leq N$) таквих да је $v_l, v_{l+1}, \dots, v_r$ *самоописујућ* низ.

Улаз

У првој линији стандардног улаза се налази један природан број N . У другој линији стандардног улаза се налазе N ненегативних целих бројева, који представљају елементе низа v .

Излаз

Исписати један број - тражени број парова индекса l, r који задовољавају својство наведено у задатку.

¹Битовски AND оператор (означен са $\&$) је бинарни оператор који пореди сваки бит два операнда и враћа вредност којој је бит постављен на 1 ако и само ако су одговарајући битови оба операнда 1. Другим речима, извршава логичко AND за сваки пар одговарајућих битова. На пример, ако имамо 4-битне бинарне бројеве $A = 1100_{(2)}$ и $B = 1010_{(2)}$, њихово битовско AND је $A \& B = 1000_{(2)}$. У C++ ова операција се означава симболом $\&$, који функционише слично као и остали оператори (на пример, $+$, $-$, итд.)

Ограничења

- $1 \leq N \leq 2\,000\,000$.
- $0 \leq v_i < 2^{60}$.

#	Посни	Ограничења
1	10	$1 \leq N \leq 200, v_i < 2^{30}$
2	3	$1 \leq N \leq 2 \cdot 10^6$, све вредности v_i су једнаке.
3	13	$1 \leq N \leq 2 \cdot 10^3, v_i < 2^{30}$
4	11	$1 \leq N \leq 4 \cdot 10^4, v_i < 2^{30}$
5	6	$1 \leq N \leq 2 \cdot 10^5, v_i = 2^k$
6	11	$1 \leq N \leq 2 \cdot 10^5, v_i < 8$
7	14	$1 \leq N \leq 2 \cdot 10^5, v_i = 2^j - 2^k$, за $0 \leq k \leq j \leq 60$
8	13	$1 \leq N \leq 2 \cdot 10^5$
9	19	Без додатних ограничења.

Примери

stdin	stdout	Објашњења
5 1 1 4 0 2	13	У овом примеру, интервали који задовољавају ово својство су: (1, 1), (1, 2), (1, 4), (1, 5), (2, 2), (2, 4), (2, 5), (3, 3), (3, 4), (3, 5), (4, 4), (4, 5), (5, 5).

Задатак Impostors

Улазна датотека `stdin`
Излазна датотека `stdout`

Постоји N соба поређаних у низ и N уљеза тако да се у почетку (у секунди $t = 0$) уљез i налази у соби број i . Из собе i постоји усмерени вентилациони отвор ка соби p_i , тако да никоја два вентилациона отвора не воде у исту собу (што чини пермутацију бројева од 1 до N). Уљези се крећу на следећи начин: уљез који се у секунди t налазио у соби i премешта се („вентује се“) у собу p_i у секунди $t + 1$.

Након K секунди зна се где се налази сваки уљез: i -ти уљез је у соби q_i . Сада се питате: колико конфигурација вентилационих отвора (пермутација p) може довести до овог стања? Пошто одговор може бити веома велик, испишите његов остатак при дељењу са $10^9 + 7$. Имајте у виду да ваш уређај за праћење догађаја може бити неисправан, па одговор може бити једнак 0.

Задатак

Напишите програм који, знајући N, K и положај сваког уљеза после K секунди, израчунава колико могућих пермутација вентилационих отвора постоји.

Улаз

Прва линија улаза садржи два цела броја N и K .

Друга линија садржи N целих бројева q_i ($1 \leq q_i \leq N$), који представљају позиције уљеза после K секунди. Гарантовано је да q формира пермутацију целих бројева од 1 до N .

Излаз

Излаз садржи један цео број: број (остатак при дељењу са $10^9 + 7$) пермутација p , таквих да ће после K секунди уљез i бити у соби q_i за свако i .

Ограничења

- $2 \leq N \leq 10^5$
- $2 \leq K \leq 10^{18}$.

#	Посни	Ограничења
1	11	$N \leq 8$ и $K \leq 20$
2	11	$N \leq 14$
3	28	$K = 2$
4	16	$N \leq 500$
5	20	$N \leq 10^4$
6	14	Без додатних ограничења.

Примери

<code>stdin</code>	<code>stdout</code>	Објашњења
3 3 1 2 3	3	Исправне пермутације p су (1, 2, 3), (2, 3, 1) и (3, 1, 2).
5 2 3 1 5 4 2	0	Не постоји одговарајућа пермутација p .