Automat

AUT

Dostępna pamięć: 128 MB.

W budynku uczelni, na której studiuje Bajtazar, stoi automat sprzedający batony. W automacie dostępnych jest n rodzajów batonów, ponumerowanych od 1 do n. Batony poszczególnych rodzajów różnią się rozmiarem oraz smakiem, więc ich ceny mogą być różne.

Niedawno Bajtazar odkrył, że automat jest popsuty. Kiedy kupi się jednego batona rodzaju i, z automatu wypada także po jednym batonie każdego z rodzajów $1, 2, \ldots, i-1$. Oczywiście tylko wtedy, gdy batony tych rodzajów są dostępne — jeśli batonów któregoś rodzaju między 1 a $i-1$ nie ma w automacie, baton tego rodzaju nie wypada. Trzeba dodać, że ta sprytna sztuczka jest możliwa do wykonania tylko wtedy, gdy w automacie faktycznie znajduje się co najmniej jeden baton i-tego rodzaju.

Bajtazar postanowił zrobić użytek z wykrytej usterki. Dysponując pewną kwotą pieniędzy, chciałby stwierdzić, jaką największą wartość (tj. sumę cen) łakoci może wydobyć z automatu za tę kwotę. Bajtazar nie musi zużyć całej dostępnej kwoty.

Wejście

Pierwszy wiersz wejścia zawiera dwie liczby całkowite n oraz k ($1 \leqslant n \leqslant 40$, $1 \leqslant k \leqslant 64\,000$), oznaczające liczbę rodzajów batonów w automacie oraz kwotę, jaką dysponuje Bajtazar. Drugi wiersz zawiera n liczb całkowitych $c_1, \ldots, c_n$ ($1 \leqslant c_i \leqslant 40$), oznaczających ceny batonów poszczególnych rodzajów. Trzeci wiersz zawiera n liczb całkowitych $l_1, \ldots, l_n$ ($0 \leqslant l_i \leqslant 40$), oznaczających liczby batonów poszczególnych rodzajów dostępnych w automacie.

Wyjście

Pierwszy i jedyny wiersz wyjścia powinien zawierać jedną liczbę całkowitą — sumę cen łakoci, jakie Bajtazar może nabyć w automacie, dysponując kwotą k.

Przykład

Dla danych wejściowych:

```
6 8
7 2 3 5 7 2
1 3 0 3 2 1
```

poprawnym wynikiem jest:

```
30
```

Wyjaśnienie do przykładu: Kupujemy batona rodzaju 6, a z automatu wypada nam także po jednym batonie rodzajów 1, 2, 4 i 5. Kupujemy batona rodzaju 4, z automatu oprócz niego wypada baton rodzaju 2.

Biuro podróży — BIU

Dostępna pamięć: 128 MB.

Bajtazar jest nauczycielem w szkole podstawowej w Bajtołach Dolnych. Korzystając z tego, że pogoda całkiem dopisuje, Bajtazar chciałby zabrać swoją klasę na wycieczkę autokarową do stolicy kraju, Bajtogrodu. Do pomocy w organizacji wycieczki Bajtazar postanowił zatrudnić biuro podróży BajTour.

Ulice w centrum Bajtogrodu tworzą regularną siatkę: każda ulica biegnie albo z zachodu na wschód, albo z południa na północ, a odległości między dwiema sąsiednimi równoległymi ulicami są równe i wynoszą jeden kilometr. Przy niektórych skrzyżowaniach znajdują się atrakcje turystyczne. Bajtoccy przewodnicy każdej atrakcji przypisali pewien współczynnik ciekawości: im wyższy współczynnik, tym dana atrakcja jest ciekawsza dla zwiedzających. Bajtazar wie, że jego podopieczni szybko się nudzą, dlatego chciałby, aby kolejno zwiedzane atrakcje miały coraz większe współczynniki ciekawości.

Biuro BajTour zgodziło się spełnić wymagania Bajtazara, ale przy tym chciałoby na wycieczce jak najwięcej zarobić. Biuro pobiera stałą stawkę jednego bajtalara za każdy kilometr trasy autokaru. Przejeżdżając między dwiema kolejnymi atrakcjami na trasie zwiedzania, autokar porusza się zawsze najkrótszą trasą biegnącą wzdłuż ulic Bajtogrodu. Ponadto, BajTour zarabia w jeszcze inny sposób: zarządcy niektórych atrakcji płacą biuru za przyprowadzanie wycieczek.

Celem BajTouru jest zaproponować trasę zgodną z warunkami postawionymi przez Bajtazara, która zagwarantuje BajTourowi możliwie największy zysk. Czy pomógłbyś w wyznaczeniu odpowiedniej trasy? Uwaga: przejechanie obok atrakcji turystycznej bez wysiadania nie liczy się jako jej zwiedzenie!

Wejście

Pierwszy wiersz wejścia zawiera dwie liczby całkowite n oraz m ($2 \leqslant n, m \leqslant 1000$), oznaczające liczbę ulic biegnących z zachodu na wschód oraz liczbę ulic biegnących z południa na północ.

Dalej następuje n wierszy zawierających opis atrakcji turystycznych. W i-tym wierszu znajduje się m liczb całkowitych $w_{i,j}$ ($0 \leqslant w_{i,j} \leqslant 10^6$) oznaczających współczynniki ciekawości atrakcji turystycznych rozmieszczonych na skrzyżowaniach i-tej ulicy biegnącej ze wschodu na zachód z kolejnymi ulicami biegnącymi z południa na północ. Współczynnik 0 oznacza brak atrakcji turystycznej, natomiast dodatnie współczynniki opisują poszczególne atrakcje. Wiadomo, że w Bajtogrodzie jest co najmniej jedna atrakcja turystyczna.

Każdy z kolejnych n wierszy zawiera po m liczb całkowitych $c_{i,j}$ ($0 \leqslant c_{i,j} \leqslant 10^9$). Liczba $c_{i,j}$, czyli j-ta liczba w i-tym z tych wierszy, oznacza kwotę (w bajtalarach), jaką biuro otrzymuje za wysłanie wycieczki do atrakcji opisanej współczynnikiem ciekawości $w_{i,j}$. Jeśli przy skrzyżowaniu nie ma atrakcji, odpowiadająca mu liczba $c_{i,j}$ jest równa 0.

Wyjście

Pierwszy i jedyny wiersz wyjścia powinien zawierać jedną liczbę całkowitą, oznaczającą (wyrażony w bajtalarach) zysk z najbardziej dochodowej dla biura trasy, odwiedzającej pewne atrakcje turystyczne w kolejności ściśle rosnących współczynników ciekawości.

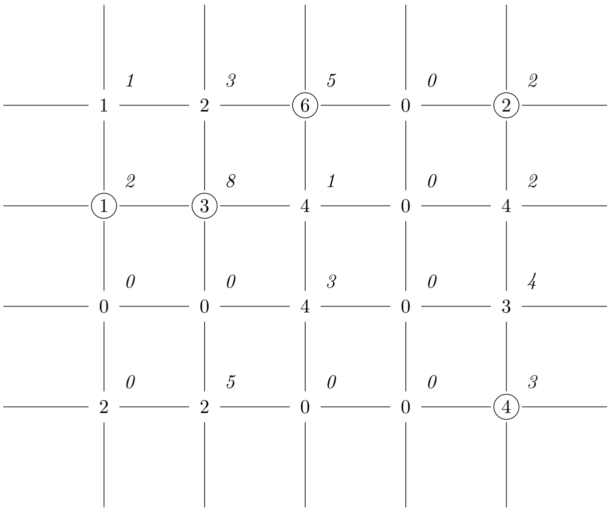
Przykład

Dla danych wejściowych:

```
4 5
1 2 6 0 2
1 3 4 0 4
0 0 4 0 3
2 2 0 0 4
1 3 5 0 2
2 8 1 0 2
0 0 3 0 4
0 5 0 0 3
```

poprawnym wynikiem jest:

```
39
```



Wyjaśnienie do przykładu: Liczby napisane na rysunku krojem prostym oznaczają współczynniki ciekawości atrakcji, a liczby napisane krojem pochyłym — dochody BajTouru za wysłanie wycieczki do poszczególnych atrakcji. Atrakcje odwiedzane na najbardziej dochodowej dla biura trasie zwiedzania są zaznaczone kółkami. Za wysłanie wycieczki do tych atrakcji biuro otrzyma, kolejno, 2, 2, 8, 3 i 5 bajtalarów. Dodatkowo, sumaryczny koszt przejazdu autokaru to 19 bajtalarów.

Ciąg CIA

Dostępna pamięć: 128 MB.

Powiemy, że ciąg liczb całkowitych $a_1, a_2, \ldots, a_n$ jest k-*parzysty*, jeśli każdy jego k-elementowy spójny fragment ma parzystą sumę.

Dla danego ciągu liczb całkowitych chcielibyśmy stwierdzić, ile minimalnie wyrazów tego ciągu musimy zmienić, aby stał się on k-parzysty.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite n oraz k ($1 \leqslant k \leqslant n \leqslant 1\,000\,000$), oznaczające długość ciągu i parametr parzystości ciągu. Drugi wiersz zawiera ciąg n liczb całkowitych $a_1, a_2, \ldots, a_n$. Każda z liczb a_i spełnia $0 \leqslant a_i \leqslant 1\,000\,000\,000$.

Wyjście

W jedynym wierszu wyjścia należy wypisać jedną liczbę całkowitą, oznaczającą najmniejszą liczbę wyrazów podanego ciągu, które trzeba zmienić, żeby ciąg stał się k-parzysty.

Przykład

Dla danych wejściowych:

```
8 3
1 2 3 4 5 6 7 8
```

poprawnym wynikiem jest:

```
3
```

natomiast dla danych:

```
8 3
2 4 2 4 2 4 2 4
```

poprawnym wynikiem jest:

```
0
```

DNA

DNA

Dostępna pamięć: 128 MB.

Szalony naukowiec Bajtazar chciałby stworzyć nowy gatunek zwierząt. W tym celu postanowił zmodyfikować kod DNA myszy bajtockiej.

Kod DNA to ciąg znaków składający się z liter `A`, `C`, `G` oraz `T`. Plan Bajtazara jest następujący. Weźmie on DNA myszy i na jego podstawie stworzy nowy kod o tej samej długości, który będzie *jak najmniej* podobny do kodu myszy. Podobieństwo dwóch kodów DNA to długość ich najdłuższego wspólnego podciągu. Najdłuższy wspólny podciąg dwóch słów x i y to najdłuższe słowo, które można uzyskać z każdego ze słów x, y przez usuwanie liter. (Zwróć uwagę, że dwa słowa mogą mieć wiele najdłuższych wspólnych podciągów, na przykład najdłuższe wspólne podciągi słów `CACCA` i `CAAC` to `CAA` oraz `CAC`.) Napisz program, który wyznaczy szukany kod DNA.

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę całkowitą n ($1 \leqslant n \leqslant 10\,000$) oznaczającą długość kodu DNA myszy bajtockiej. W drugim wierszu znajduje się kod DNA myszy w postaci ciągu n wielkich liter należących do zbioru $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$.

Wyjście

Pierwszy wiersz wyjścia powinien zawierać jedną liczbę całkowitą — podobieństwo kodu myszy bajtockiej oraz kodu znalezionego przez Twój program. W drugim wierszu należy wypisać ciąg składający się z n liter `A`, `C`, `G` lub `T`. Powinien być to kod DNA, który jest jak najmniej podobny do kodu podanego na wejściu. Jeśli istnieje wiele poprawnych odpowiedzi, Twój program może wypisać dowolną z nich.

Przykład

Dla danych wejściowych:

```
4
GACT
```

jednym z poprawnych wyników jest:

```
1
TCAG
```

EWALUACJA EWA

Dostępna pamięć: 128 MB.

Dane jest wyrażenie matematyczne E, w którym występują: stałe od 0 do 9, zmienne od a do z, a także operacje: dodawania, mnożenia i potęgowania o stałym wykładniku. Co ciekawe, każda ze zmiennych $a, b, \ldots, z$ występuje w wyrażeniu E co najwyżej raz. Zastanawiamy się, dla danej liczby pierwszej p, ile pierwiastków modulo p ma wielomian wyznaczony przez to wyrażenie. Innymi słowy, pytamy, ile jest podstawień liczb od 0 do $p-1$ pod zmienne występujące w E, dla których wartość wyrażenia E jest podzielna przez p. Ponieważ szukana liczba pierwiastków może być duża, wystarczy nam reszta z jej dzielenia przez 30 011.

Przykładowo, wielomian reprezentowany przez wyrażenie

$$E = ((a+y)\cdot(z+8))^2$$

ma 15 pierwiastków modulo $p=3$, m.in. następujące trzy pierwiastki:

$$(a=0,\ y=0,\ z=0), \qquad (a=1,\ y=2,\ z=0), \qquad (a=2,\ y=0,\ z=1).$$

Formalnie, *wyrażenie* definiujemy następująco:

- Każda stała `0`, `1`, ..., `9` jest wyrażeniem.
- Każda zmienna `a`, `b`, ..., `z` jest wyrażeniem.
- Jeśli `A` i `B` są dowolnymi wyrażeniami, to wyrażeniami są także `(A+B)` i `(A*B)`. Pierwsze z nich oznacza sumę wyrażeń `A` i `B`, zaś drugie — ich iloczyn.
- Jeśli `A` jest dowolnym wyrażeniem, a `B` jest stałą z zakresu `2`, `3`, ..., `9`, to wyrażeniem jest także `(A^B)` (wyrażenie `A` do potęgi `B`).

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę pierwszą p ($2 \leqslant p < 15\,000$). Drugi wiersz zawiera wyrażenie E zgodne z podaną specyfikacją, opisane przez ciąg złożony z co najwyżej 300 znaków `0`, `1`, ..., `9`, `a`, `b`, ..., `z`, `+`, `*`, `^`, `(`, `)`. W podanym ciągu nie występują odstępy.

Wyjście

Oznaczmy przez k liczbę pierwiastków modulo p wielomianu E. Twój program powinien wypisać jedną nieujemną liczbę całkowitą: resztę z dzielenia k przez 30 011.

Przykład

Dla danych wejściowych:

```
3
(((a+y)*(z+8))^2)
```

poprawnym wynikiem jest:

```
15
```

Formuła 1 FOR

Dostępna pamięć: 128 MB.

Mały Robert Kubita bardzo lubi oglądać wyścigi Formuły 1, które w Bajtocji odbywają się na torze prowadzącym z Bajtogrodu do Bitowic. Najbardziej ekscytującymi dla Roberta momentami wyścigu są manewry wyprzedzania, dlatego chłopiec chciałby ich widzieć jak najwięcej.

Marzy mu się zobaczyć wyścig, który spełniłby następujące założenia: ścigałoby się w nim n bolidów, a dla każdego i ($1 \leqslant i \leqslant n$) bolid, który startował z i-tej pozycji, wykonałby podczas wyścigu a_i manewrów wyprzedzania. Zakładamy, że w każdej chwili wyścigu odbywa się co najwyżej jeden manewr wyprzedzania, który polega na tym, że pewien bolid przesuwa się przed bolid bezpośrednio poprzedzający go.

Robert zastanawia się, czy taki wyścig jest w ogóle możliwy. Poprosił Cię o pomoc w rozstrzygnięciu tej kwestii.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita t, oznaczająca liczbę zestawów testowych opisanych w dalszej części wejścia.

Opis każdego zestawu składa się z dwóch wierszy. W pierwszym z nich znajduje się jedna liczba całkowita n ($1 \leqslant n \leqslant 1\,000\,000$), oznaczająca liczbę bolidów biorących udział w wyścigu. W drugim wierszu znajduje się ciąg n liczb całkowitych $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 10^9$), który zadaje, ile manewrów wyprzedzania muszą wykonać poszczególne bolidy.

Rozmiar żadnego pliku wejściowego nie przekracza 20 MB.

Wyjście

Twój program powinien wypisać t wierszy z odpowiedziami dla kolejnych zestawów testowych. Odpowiedzią dla zestawu jest słowo `TAK` albo `NIE`, w zależności od tego, czy da się zrealizować wyścig zgodnie z wytycznymi Roberta.

Przykład

Dla danych wejściowych:

```
3
2
0 1
3
0 1 4
3
1 1 3
```

poprawnym wynikiem jest:

```
TAK
NIE
TAK
```

Generał kontra gawiedź GEN

Dostępna pamięć: 128 MB.

W Bajtogrodzie niesamowite poruszenie! Bajtoccy archeolodzy odkryli w pobliżu miasta szczątki dinozaurów. Gdy tylko gawiedź się o tym zwiedziała, ten i ów jął wybierać się na wykopalisko po to, by zwędzić mniejszą lub większą kość. Proceder ten stał się tak nagminny, że postanowiono do ochrony terenu wykopalisk zatrudnić wojsko.

Generał Bajtazar rozlokował n żołnierzy w *strategicznych punktach* na terenie wykopalisk. (Żołnierze nie mogą stanąć gdzie bądź, aby nie przeszkadzać archeologom. Poza tym muszą mieć też dobrą widoczność, aby chronić teren wykopalisk.) Powiemy, że dany punkt terenu jest *chroniony*, jeśli ruszając się z niego w jakimkolwiek kierunku, zbliżymy się do któregoś z żołnierzy (tj. nasza odległość od tego żołnierza zmaleje).

Bajtazarowi przydzielono właśnie nowego rekruta. Generał może go umieścić w jednym z m jeszcze nieobsadzonych punktów strategicznych. Dla każdego z tych punktów interesuje go, jaka będzie powierzchnia chronionego terenu, gdy nowy żołnierz stanie w tym punkcie.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite n i m ($3 \leqslant n \leqslant 100\,000$, $1 \leqslant m \leqslant 100\,000$), oznaczające liczbę rozstawionych żołnierzy i liczbę nieobsadzonych punktów strategicznych. Kolejne n wierszy opisuje żołnierzy: w i-tym z tych wierszy znajdują się liczby całkowite x_i, y_i ($-10^8 \leqslant x_i, y_i \leqslant 10^8$), które oznaczają współrzędne punktu (w prostokątnym układzie współrzędnych), w którym stoi i-ty żołnierz. W kolejnych m wierszach zapisane są w tym samym formacie kolejne nieobsadzone punkty strategiczne. Punkty podane na wejściu nie powtarzają się.

Można założyć, że powierzchnia terenu chronionego przez już rozstawionych żołnierzy jest dodatnia.

Wyjście

Na wyjście należy wypisać dokładnie m wierszy. W i-tym wierszu należy wypisać całkowitą powierzchnię chronionego terenu, jeśli nowy rekrut stanie w i-tym nieobsadzonym punkcie strategicznym. Liczby należy wypisywać z dokładnie jedną cyfrą po kropce dziesiętnej.

Przykład

Dla danych wejściowych:

```
3 2
0 0
2 -1
1 2
3 1
1 0
```

poprawnym wynikiem jest:

```
5.0
2.5
```

Hydra HYD

Dostępna pamięć: 128 MB.

Bituś dostał na urodziny grę komputerową o nazwie *Niesamowite przygody Rycerza Bajtazara*. Zabawa polega na kierowaniu postacią tytułowego rycerza, który przemierza królestwo Bajtocji i pomaga uciśnionym przez poczwary, poborców podatkowych i gradobicia. Bituś przeszedł już prawie całą grę, ale utknął na ostatnim poziomie, w którym Bajtazar walczy z wielkim morskim wężem — Bajtocką Hydrą.

Do walki z potworem Bajtazar używa swojego miecza. W grze dostępne są dwa rodzaje ciosów: rycerz może albo uciąć głowę węża, albo (co oczywiście wymaga więcej wysiłku) zmasakrować tę głowę. Jednakowoż ucięcie głowy węża, choć prostsze, powoduje, że w miejscu odcięcia z szyi węża odrastają nowe głowy. Wodny potwór zostanie pokonany dopiero wtedy, gdy Bajtazar pozbawi go wszystkich głów i żadna głowa nie będzie już mogła odrosnąć.

Bajtocka Hydra może mieć n rodzajów głów, które będziemy oznaczali liczbami od 1 do n. Na samym początku wąż ma jedną głowę rodzaju 1. Głowa rodzaju i (dla $1 \leqslant i \leqslant n$) charakteryzuje się następującymi cechami: liczbą machnięć miecza u_i, które musi wykonać Bajtazar, aby uciąć tę głowę, liczbą machnięć miecza z_i, która jest wymagana do zmasakrowania tej głowy, oraz listą r_i rodzajów głów $g_{i,1}, \ldots, g_{i,r_i}$, które odrastają na miejsce głowy rodzaju i po jej ucięciu.

Podpowiedz Bitusiowi, ile minimalnie machnięć mieczem należy wykonać, aby pokonać Hydrę.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \leqslant n \leqslant 200\,000$), oznaczająca liczbę rodzajów głów Hydry. Kolejne n wierszy opisuje poszczególne rodzaje głów; w i-tym z tych wierszy opisana jest głowa rodzaju i. Wiersz ten zaczyna się trzema liczbami całkowitymi u_i, z_i, r_i ($1 \leqslant u_i < z_i \leqslant 10^9$, $1 \leqslant r_i$), po których następują liczby całkowite $g_{i,1}, \ldots, g_{i,r_i}$ ($1 \leqslant g_{i,j} \leqslant n$). Suma liczb r_i nie przekracza 1 000 000.

Wyjście

W jedynym wierszu wyjścia należy wypisać minimalną liczbę machnięć mieczem, która jest potrzebna, by wygrać grę.

Przykład

Dla danych wejściowych:

```
4
4 27 3 2 3 2
3 5 1 2
1 13 2 4 2
5 6 1 2
```

poprawnym wynikiem jest:

```
26
```

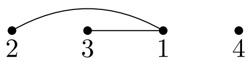
Inwersje

INW

Dostępna pamięć: 128 MB.

Bajtazar odkrył nową rodzinę grafów nieskierowanych, które można reprezentować za pomocą inwersji. Niech $V = \{1, 2, \ldots, n\}$ będzie zbiorem wierzchołków grafu, natomiast $a_1, a_2, \ldots, a_n$ — pewnym ciągiem parami różnych liczb ze zbioru V. Wierzchołki a_i oraz a_j są połączone krawędzią w grafie, jeśli para (i, j) jest *inwersją* w tym ciągu, to znaczy $i < j$ oraz $a_i > a_j$.

Dla przykładu rozważmy $n = 4$ i ciąg $2, 3, 1, 4$. Wtedy uzyskujemy graf jak na rysunku:



Bajtazar chce pokazać, że wymyślona przez niego reprezentacja jest użyteczna. Postanowił napisać program, który wyznacza *spójne składowe* grafu. Przypomnijmy, że dwa wierzchołki $u, v \in V$ znajdują się w tej samej spójnej składowej grafu, jeśli istnieje taki ciąg wierzchołków, którego pierwszym wyrazem jest u, ostatnim — v, a każde dwa kolejne wierzchołki są połączone krawędzią grafu. W naszym przykładzie mamy dwie spójne składowe: $\{1, 2, 3\}$ oraz $\{4\}$.

Pomóż Bajtazarowi!

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \leqslant n \leqslant 1\,000\,000$) oznaczająca liczbę wierzchołków grafu. W drugim wierszu znajduje się ciąg n liczb całkowitych $a_1, a_2, \ldots, a_n$.

Wyjście

W pierwszym wierszu wyjścia należy wypisać liczbę spójnych składowych grafu; oznaczmy tę liczbę przez m. W każdym z kolejnych m wierszy należy podać opis jednej spójnej składowej grafu. Na początku wiersza wypisać należy liczbę k oznaczającą rozmiar składowej, a następnie *rosnący* ciąg k numerów wierzchołków tej składowej. Składowe należy wypisać w takiej kolejności, by pierwsze numery wierzchołków z każdego wiersza tworzyły ciąg rosnący. Innymi słowy, jeśli S i S' są dwiema składowymi, $u \in S$, $v \in S'$ są ich najmniejszymi wierzchołkami oraz $u < v$, to składową S należy wypisać przed składową S'.

Przykład

Dla danych wejściowych:

```
4
2 3 1 4
```

poprawnym wynikiem jest:

```
2
3 1 2 3
1 4
```

Jutro

JUT

Dostępna pamięć: 128 MB.

prokrastynacja (łac. *procrastinatio* z *pro cras* — na jutro)
— tendencja do nieustannego przekładania pewnych czynności na później

Bajtazar ma w zwyczaju odkładać wszystko na ostatnią chwilę. Właściwie można powiedzieć, że prokrastynacja to jego drugie imię. Niemniej jednak, jeśli się do czegoś zobowiąże, to można na niego liczyć. Bajtazar wstał dzisiaj rano i wypisał sobie listę n zadań, które musi wykonać w najbliższym czasie. Wykonanie i-tego zadania z listy zajmie mu dokładnie d_i kolejnych dni, zaś musi ono zostać ukończone przed upływem t_i dni, licząc od dziś. Bajtazar chciałby wiedzieć, jak długo może zwlekać, zanim będzie musiał wziąć się w końcu do roboty. Pomóż mu i napisz program, który to obliczy. Bajtazar mógłby to zrobić sam, ale byłoby to wbrew jego naturze.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \leqslant n \leqslant 1\,000\,000$), oznaczająca liczbę zadań, które musi zrealizować Bajtazar. W kolejnych n wierszach znajdują się opisy zadań: i-ty z tych wierszy zawiera dwie liczby całkowite d_i i t_i ($1 \leqslant d_i, t_i \leqslant 10^9$). Zakładamy, że Bajtazar będzie w stanie zrealizować wszystkie zaplanowane zadania.

Wyjście

Na wyjście należy wypisać jedną liczbę całkowitą k, oznaczającą liczbę dni, przez które Bajtazar może unikać pracy. Innymi słowy, najpóźniej w dniu $k+1$ musi zacząć wykonywać jakieś zadanie, aby był w stanie zrealizować swój plan.

Przykład

Dla danych wejściowych:

```
3
2 8
1 13
3 10
```

poprawnym wynikiem jest:

```
5
```

Wyjaśnienie do przykładu: Bajtazar przez pięć dni odpoczywa, przez następne pięć dni wykonuje zadanie pierwsze oraz trzecie (w tej kolejności), a następnie jeden z trzech kolejnych dni poświęca na wykonanie zadania drugiego.

Króliki KRO

Dostępna pamięć: 256 MB.

Bajtazar postanowił zacząć żyć ekologicznie, i został hodowcą sałaty. Jak wiadomo, bajtockie króliki uwielbiają sałatę, nic więc dziwnego, że wkrótce zawitały one do ogródka Bajtazara.

Wokół domu Bajtazara rozmieszczone jest n grządek sałaty, które będziemy numerować od 1 do n. Każde dwie kolejne grządki sąsiadują ze sobą, tzn. dla każdego $i = 1, 2, \ldots, n-1$ grządki o numerach i oraz $i+1$ sąsiadują ze sobą. Dodatkowo, grządka o numerze n sąsiaduje z grządką o numerze 1. Na grządce o numerze i siedzi a_i królików i pałaszuje sałatę Bajtazara.

Bajtazar chciałby przegonić z ogródka jak najwięcej królików. Ma do dyspozycji swoją starą, wierną strzelbę, w której jest k nabojów. Króliki są bardzo płochliwe, więc gdy tylko Bajtazar wystrzeli w kierunku grządki o numerze i, to wszystkie króliki z tej grządki uciekają w siną dal. Co więcej, przestraszone króliki z obu sąsiednich grządek przeskakują na dalsze grządki, tzn. kicają na grządkę sąsiadującą z tą, na której stoją — oczywiście nie tą, w kierunku której strzelał Bajtazar.

Pomóż Bajtazarowi i wyznacz maksymalną liczbę królików, które może on wygonić z ogródka za pomocą co najwyżej k strzałów.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby całkowite n i k ($5 \leqslant n \leqslant 2000$, $1 \leqslant k \leqslant n$), oznaczające liczbę grządek w ogródku i liczbę nabojów w strzelbie Bajtazara. W drugim wierszu znajduje się ciąg liczb całkowitych $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 1\,000\,000$), które oznaczają liczby królików na kolejnych grządkach.

Wyjście

Na wyjście należy wypisać jedną liczbę całkowitą — maksymalną liczbę królików, które Bajtazar jest w stanie przegonić z ogródka.

Przykład

Dla danych wejściowych:

```
5 2
6 1 5 3 4
```

poprawnym wynikiem jest:

```
13
```

Wyjaśnienie do przykładu: Bajtazar przegania 6 królików z grządki numer 1 (wskutek czego króliki z grządki numer 5 przeskakują na grządkę numer 4, zaś króliki z grządki numer 2 przeskakują na grządkę numer 3), a następnie przegania 7 królików z grządki numer 4.