Kaucja (A)

Limit pamięci: 256 MB Limit czasu: 1.00 s

Jaś zbiera opakowania kaucyjne po produktach spożywczych. Zebrał już całkiem pokaźny ich zbiór, teraz zastanawia się, jak dużo kaucji jest w stanie odzyskać.

Są 3 typy opakowań, które podlegają kaucji:

- butelki plastikowe (`P`) – podlegające kaucji 0,50 zł
- puszki metalowe (`M`) – podlegające kaucji 0,50 zł
- butelki szklane (`S`) – podlegające kaucji 1,00 zł

Wejście

W pierwszym wierszu wejścia znajduje się liczba N określająca liczbę opakowań, jakie posiada Jaś. W drugim (i ostatnim) wierszu wejścia znajduje się ciąg N znaków ze zbioru {`P`, `M`, `S`} oddzielonych spacjami, oznaczających odpowiednio butelkę plastikową, puszkę metalową lub butelkę szklaną.

Wyjście

W pierwszym (i jedynym) wierszu wyjścia powinna się znaleźć sumaryczna kwota (w złotówkach), jaką otrzyma Jaś za zwrot wszystkich posiadanych opakowań. Kwota powinna być wypisana w postaci liczby zmiennoprzecinkowej, z kropką jako separatorem dziesiętnym, tak jak w przykładach. Należy ją wypisać z dokładnością do dokładnie dwóch miejsc po przecinku.

Ograniczenia

$1 \leq N \leq 10^6$.

Przykłady

Wejście	Wyjście	Wyjaśnienie
`3` `P P P`	`1.50`	Jaś ma 3 butelki plastikowe, kaucja za nie wynosi $1,5$ zł.

Wejście	Wyjście	Wyjaśnienie
`3` `P M S`	`2.00`	Jaś ma po jednej butelce plastikowej, puszce metalowej oraz butelce szklanej. Kaucja za nie wynosi 2 zł.

Wejście	Wyjście	Wyjaśnienie
`1` `S`	`1.00`	Jaś ma tylko jedną butelkę szklaną. Kaucja wynosi 1 zł.

Kółko i krzyżyk (B)

Limit pamięci: 256 MB Limit czasu: 1.00 s

Bajtazar grał ze swoim kolegą w klasyczną grę w kółko i krzyżyk na planszy o wymiarach 3 × 3. Grę zawsze rozpoczynał gracz X, a obaj gracze wykonywali ruchy naprzemiennie (`X`, potem `O`, potem `X` itd.). Partia dobiegła końca i jej ostateczny stan został zapisany w pliku tekstowym.

Twoim zadaniem jest napisanie programu, który wczyta planszę, przeanalizuje jej stan i wskaże wynik zakończonej rozgrywki.

Gracz wygrywa, jeśli jako pierwszy ułoży trzy swoje symbole (`X` lub `O`) w jednej linii: poziomo, pionowo lub po przekątnej. W takim przypadku gra natychmiast się kończy. Jeśli po wykonaniu 9 ruchów cała plansza zostanie zapełniona, a żaden gracz nie utworzył linii trzech swoich symboli, mecz kończy się remisem.

Wejście

Wejście składa się z 3 wierszy. Każdy wiersz zawiera dokładnie 3 znaki spośród następujących:

- `X` — pole zaznaczone przez gracza X,
- `O` — pole zaznaczone przez gracza O,
- `.` — pole puste.

Możesz założyć, że stan planszy jest poprawny, to znaczy przedstawia ostateczny stan w pełni rozegranej partii w kółko i krzyżyk, którą rozpoczynał gracz X. Gwarantowane jest, że nie wystąpi stan gry przerwanej przed zakończeniem ani sytuacja, w której wygraliby obaj gracze naraz.

Wyjście

Wypisz jeden znak:

- `X` — jeśli wygrał gracz X,
- `O` — jeśli wygrał gracz O,
- `R` — jeśli gra zakończyła się remisem.

Przykład

Wejście	Wyjście

`XXX`
`OO.`
`...`
	`X`

Trójkąty (C)

Limit pamięci: 256 MB Limit czasu: 1.00 s

Jak by to było być trójkątem? To pytanie, które niewiele osób sobie zadaje. My spróbujemy odpowiedzieć na inne pytanie: Czy trójkąt równoboczny da się podzielić na dokładnie N trójkątów równobocznych?

Uwagi:

- Trójkąty w podziale nie muszą być tego samego rozmiaru.
- Trójkąty w podziale nie mogą na siebie nachodzić – mogą się jedynie stykać bokami i wierzchołkami.
- Trójkąty mają pokrywać całą powierzchnię oryginalnego trójkąta.

Wejście

W pierwszym i jedynym wierszu wejścia znajduje się liczba całkowita N.

Wyjście

W pierwszym i jedynym wierszu wyjścia powinno się znaleźć pojedyncze słowo `TAK`, jeśli trójkąt równoboczny da się podzielić na N trójkątów równobocznych, lub `NIE`, jeśli nie da się dokonać takiego podziału.

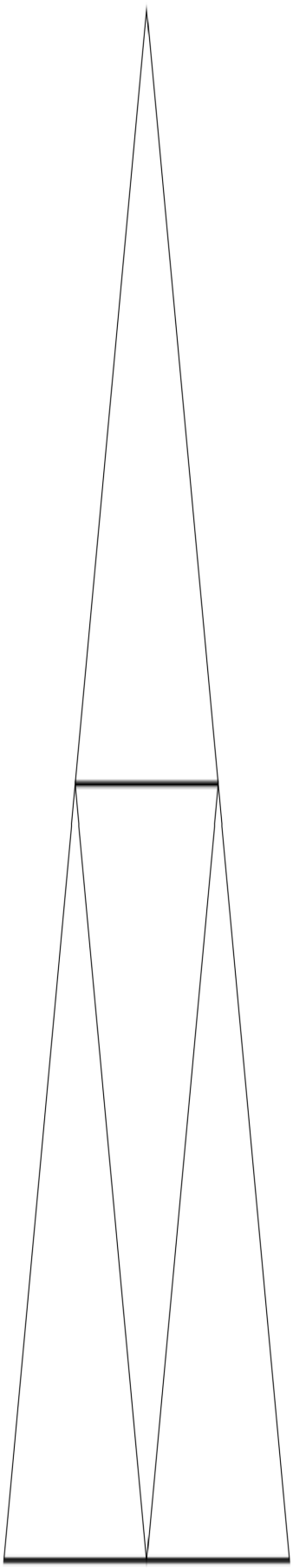
Ograniczenia

$1 \leq N \leq 100$.

Przykłady

Wejście	Wyjście	Wyjaśnienie
`2`	`NIE`	Nie da się podzielić trójkąta równobocznego na 2 trójkąty równoboczne.

Wejście	Wyjście	Wyjaśnienie
`4`	`TAK`	Trójkąt równoboczny da się podzielić na 4 trójkąty równoboczne, co ilustruje rysunek.



Podział trójkąta równobocznego na 4 trójkąty równoboczne

Wejście	**Wyjście**
100	TAK

Kolorowanie Alfabetu (D)

Limit pamięci: 256 MB Limit czasu: 1.00 s

Jaś niedawno zaczął chodzić do szkoły. Jego ulubionymi zajęciami są kolorowanie oraz uczenie się alfabetu. Jaś uznał, że fantastycznym pomysłem byłoby połączenie tych dwóch pasji i pokolorowanie alfabetu.

Jaś napisał sobie pewne słowo (bardzo długie i niekoniecznie będące prawdziwym słowem). Postanowił, że dla każdej litery alfabetu pokoloruje wszystkie wystąpienia tej litery w swoim słowie na ten sam kolor. Jaś zastanawia się, jak bardzo kolorowe może być słowo po pokolorowaniu. *Kolorowość* słowa rozumie jako liczbę jednokolorowych bloków danego pokolorowanego słowa, gdzie blok to maksymalny przedział liter o tym samym kolorze. Pomóż mu – podaj maksymalną osiągalną kolorowość słowa Jasia.

Wejście

W pierwszym wierszu wejścia znajduje się liczba naturalna N. W drugim i ostatnim wierszu wejścia znajduje się słowo składające się z N znaków – słowo Jasia. **W słowie tym występuje tylko 12 pierwszych liter alfabetu:** `A`**,** `B`**,** `C`**,** `D`**,** `E`**,** `F`**,** `G`**,** `H`**,** `I`**,** `J`**,** `K`**,** `L` (Jaś jeszcze nie nauczył się innych). Wszystkie litery na wejściu są wielkie (Jaś jeszcze nie nauczył się pisowni małych).

Wyjście

W pierwszym i jedynym wierszu wyjścia powinna się znaleźć pojedyncza liczba naturalna – maksymalna możliwa kolorowość słowa Jasia po kolorowaniu. Jaś ma przy sobie tylko 3 kredki: czerwoną, zieloną i niebieską, więc **rozważamy jedynie kolorowania korzystające z maksymalnie 3 kolorów**.

Ograniczenia

$1 \leq N \leq 10^6$.

Przykłady

Wejście	Wyjście	Wyjaśnienie
`10` `AAAAAAAAAA`	`1`	Dowolne kolorowanie alfabetu skutkuje pokolorowaniem całego słowa na jeden kolor, więc maksymalna kolorowość wynosi 1.

Wejście	Wyjście	Wyjaśnienie
`12` `AAAABBBCCCCC`	`3`	Maksymalna kolorowość wynosi 3. Realizuje ją np. kolorowanie `A` na czerwono, `B` na zielono i `C` na niebiesko. Kolor pozostałych liter nie ma znaczenia.

Wejście	Wyjście	Wyjaśnienie
`12` `ABCDEFABCDEF`	`12`	Możemy pokolorować alfabet następująco: litery `A` i `D` na czerwono, `B` i `E` na zielono, `C` i `F` na niebiesko, a pozostałe litery dowolnie, otrzymując maksymalną możliwą kolorowość, czyli 12.

Wejście	Wyjście	Wyjaśnienie
`12` `ABCDBCADCADB`	`11`	Maksymalna kolorowość wynosi 11.

Ocenianie

Podzadanie	Dodatkowe warunki	Punkty
1	$N \leq 100$	50
2	Brak dodatkowych ograniczeń	50

Maratończyk (E)

Limit pamięci: 256 MB Limit czasu: 1.00 s

Pewien biegacz postanowił wziąć udział w morderczym ultramaratonie składającym się z N kolejnych odcinków trasy. Pokonanie i-tego odcinka o własnych siłach wymaga zużycia dokładnie A_i jednostek energii.

Przed startem maratończyk ma w pełni naładowany zapas energii równy X. W trakcie zawodów może co najwyżej K razy skorzystać ze specjalnej taksówki dla biegaczy.

Skorzystanie z taksówki na danym odcinku niesie ze sobą dwie korzyści: 1. Biegacz nie zużywa na nim ani jednej jednostki energii. 2. Podczas jazdy odpoczywa, dzięki czemu po opuszczeniu pojazdu poziom jego energii wynosi ponownie dokładnie X (niezależnie od tego, ile energii zostało mu przed wejściem).

Pomóż biegaczowi obliczyć minimalną początkową pojemność energii X, która pozwoli mu ukończyć całą trasę tak, aby w żadnym momencie poziom energii nie spadł poniżej zera.

Wejście

W Pierwszy wierszu wejścia znajdują się dwie liczby całkowite N oraz K oznaczające odpowiednio liczbę odcinków maratonu oraz maksymalną liczbę dozwolonych przejazdów taksówką.

Drugi wiersz wejścia zawiera N liczb całkowitych $A_1, A_2, \ldots, A_N$ ($1 \leq A_i \leq 10^9$) pooddzielanych pojedynczymi odstępami, określających koszt energetyczny pokonania kolejnych odcinków.

Wyjście

Twój program powinien wypisać jedną liczbę całkowitą — minimalny początkowy poziom energii X, z którym maratończyk może ukończyć bieg.

Ograniczenia

$1 \leq N \leq 10^5$, $0 \leq K \leq N$, $1 \leq A_i \leq 10^9$

Przykład

Wejście
```
5 2
5 8 3 7 4
```

Wyjście
```
5
```

Wyjaśnienie

1. Startuje z energią 5. Po przebiegnięciu zostaje mu $5 - 5 = 0$.
2. Bierze taksówkę (1. użycie). Wysiada z pełną energią 5.
3. Biegnie o własnych siłach; zostaje mu $5 - 3 = 2$.
4. Bierze taksówkę (2. użycie). Wysiada z energią 5.
5. Biegnie o własnych siłach; zostaje mu $5 - 4 = 1 \geq 0$.

Wejście
```
4 0
4 2 3 5
```

Wyjście
```
14
```

Ocenianie

Podzadanie	Warunki	Punkty
1	$K = 1$	15
2	$N \leq 20$	15

Podzadanie	Warunki	Punkty
3	$N \leq 1000$	30
4	Brak dodatkowych ograniczeń	40

Zderzające kulki (F)

Limit pamięci: 256 MB Limit czasu: 1.00 s

W wąskiej rurze leży w jednej linii N kulek. Kolejne kulki (patrząc od lewej do prawej) mają początkowe masy $M_1, M_2, \ldots, M_N$.

Uwaga fizyczna: Masy te mogą przyjmować wartości dodatnie, zerowe, a nawet **ujemne**. Obiekty te składają się z niestabilnej materii kwantowej – w trakcie gwałtownych zderzeń kulka może stracić więcej masy, niż faktycznie posiadała, co powoduje „przejście przez zero" i uzyskanie ujemnej masy (działającej jak egzotyczna antywaga odpychająca otoczenie).

W jednym ruchu możesz wybrać dwie sąsiadujące ze sobą kulki i zamienić je miejscami. Zamiana wymaga jednak zderzenia, które zmienia ich masy: Lewa kulka przeskakująca w prawo traci 1 jednostkę masy, natomiast prawa kulka przeskakująca w lewo zyskuje 1 jednostkę masy.

Możesz wykonać dowolną liczbę takich ruchów (masy kulek mogą w trakcie lub na końcu stać się zerowe albo ujemne). Twoim zadaniem jest sprawdzenie, czy da się ustawić kulki tak, aby ich ostateczne masy tworzyły ciąg niemalejący (czyli dla każdego i od 1 do $N-1$, masa kulki na pozycji i ma być mniejsza lub równa masie kulki na pozycji $i+1$).

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita N oznaczająca liczbę kulek.

Drugi wiersz wejścia zawiera N liczb całkowitych $M_1, M_2, \ldots, M_N$ pooddzielanych pojedynczymi odstępami, określających początkowe masy kolejnych kulek.

Wyjście

Twój program powinien wypisać jedno słowo — `TAK`, jeśli da się ustawić kulki w ciąg niemalejący, lub `NIE` w przeciwnym wypadku.

Ograniczenia

$1 \leq N \leq 200\,000$, $1 \leq M_i \leq 10^9$

Przykład

Wejście
```
3
5 2 4
```

Wyjście
```
TAK
```

Wyjaśnienie

1. Początkowe masy kulek to $[5, 2, 4]$.
2. Zamieniamy pierwszą i drugą kulkę miejscami: pierwsza traci 1 jednostkę masy ($5 - 1 = 4$), a druga zyskuje 1 jednostkę ($2 + 1 = 3$).
3. Otrzymujemy układ $[3, 4, 4]$, który jest ciągiem niemalejącym ($3 \leq 4 \leq 4$).

Wejście
```
2
5 4
```

Wyjście
```
NIE
```

Ocenianie

Podzadanie	Warunki	Punkty
1	$N \leq 10$	20

Podzadanie	Warunki	Punkty
2	$N \leq 1000$	30
3	Brak dodatkowych ograniczeń	50

Liczby Chropowate (G)

Limit pamięci: 256 MB Limit czasu: 5.00 s

Bajtek uważa, że liczby pierwsze są chropowate, natomiast Bitek uważa, że chropowate są liczby postaci $6p^k + 1$, gdzie p jest liczbą pierwszą, a k jest liczbą całkowitą dodatnią. Ta różnica zdań jest źródłem wielu nieporozumień i kłótni.

Ostatnio Bajtek i Bitek zastanawiają się, jak często zgadzają się co do chropowatości liczb. Twoim zadaniem jest pomóc im to rozstrzygnąć.

Wejście

W pierwszym i jedynym wierszu wejścia znajdują się dwie liczby naturalne: A i B oddzielone spacją.

Wyjście

W pierwszym i jedynym wierszu wyjścia powinna się znaleźć pojedyncza liczba całkowita – liczba takich liczb całkowitych x w przedziale $[A, B]$, że Bitek i Bajtek zgadzają się co do tego, czy x jest chropowata. (Czyli albo obaj uważają, że jest chropowata, albo obaj uważają, że nie jest.)

Ograniczenia

$1 \leq A \leq B \leq 10^{16}$, $B - A < 10^6$.

Przykłady

Wejście	Wyjście	Wyjaśnienie
`67 67`	`1`	$67 = 6 \cdot 11^1 + 1$ jest zarówno liczbą pierwszą (chropowatą według Bajtka), jak i liczbą postaci $6p^k + 1$ (chropowatą według Bitka).

Wejście	Wyjście	Wyjaśnienie
`10 25`	`12`	W przedziale $[10, 25]$ znajduje się 5 liczb pierwszych: 11, 13, 17, 19 i 23, natomiast chropowate według Bitka są liczby: 13, 19, 25. Zatem Bitek i Bajtek zgodnie określają chropowatość 12 liczb: 10, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 24.

Wejście	Wyjście
`9999999999000001 1000000000968701`	`968701`

Ocenianie

Podzadanie	Dodatkowe warunki	Punkty
1	$A = B$	40
2	$B - A < 100$	40
3	Brak dodatkowych ograniczeń	20