

1. Number of islands (25 punktów)

Zadanie

Dana jest tablica G , składająca się z m stringów o długości n każdy. Stringi zawierają wyłącznie znaki 's' (sea) lub 'l' (land).

Pojedynczy ruch polega na przemieszczeniu się z danego pola będącego lądem na sąsiednie (w czterech kierunkach) pole, również oznaczone jako ląd, lub wyjście poza granicę obszaru (tylko jeżeli bieżące pole leży na brzegu).

Napisz program, który wyliczy z ilu pól oznaczonych jako ląd nie można wyjść poza granicę obszaru w dowolnej liczbie kroków.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite, m (liczba stringów) i n (liczba znaków w każdym stringu). W każdej z kolejnych m linii znajduje się string o długości n i składający się z liter 's' i 'l', opisujący rodzaje pól w danym wierszu gridu.

Ograniczenia

1. $1 \leq m, n \leq 500$
2. $G[i][j] = \text{'s'}$ lub $G[i][j] = \text{'l'}$; $i = 0, \dots, m - 1, j = 0, \dots, n - 1$

Wyjście

W pierwszym i jedynym wierszu program powinien wypisać jedną liczbę całkowitą - liczbę pól oznaczonych jako ląd, z których nie można wyjść poza granicę obszaru w dowolnej liczbie kroków.

Przykład

Dla danych wejściowych:

```
4 4
slls
ssls
slss
ssss
```

poprawną odpowiedzią jest:

```
1
```

W gridzie znajduje się 1 pole (o współrzędnych (2, 1)) będące lądem i otoczone ze wszystkich czterech stron morzem. Z pozostałych trzech pól "lądowych" można dojść do granicy obszaru "suchą nogą".

2. Square flips (25 punktów)

Zadanie

Mamy n kwadratowych klocków pomalowanych z jednej strony na biało a z drugiej na czarno. Układamy je w rzędzie, przy czym kolor górnej powierzchni i -tego klocka jest opisany i -tym znakiem stringu S . String S składa się z dokładnie n znaków 'b' (black) lub 'w' (white). Jeżeli $S[i] = 'b'$ to i -ty klocek jest ułożony czarną stroną do góry, jeżeli $S[i] = 'w'$ to odwracamy i -ty kwadrat białą stroną.

Dodatkowo dana jest liczba całkowita K .

W danym kroku możemy wybrać K kolejnych klocków i odwrócić je na drugą stronę (czyli z czarnej na białą lub z białej na czarną).

Napisz program, który wyznaczy minimalną liczbę kroków, po wykonaniu których wszystkie kwadraty będą odwrócone białą stroną. Jeżeli jest to niemożliwe program zwraca -1.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite, n (długość stringu S) i K (liczba klocków, które odwracamy w jednym kroku). W kolejnej linii znajduje się string S składający się z n liter 'b' i 'w', opisujący kolory widoczne na kolejnych klockach.

Ograniczenia

1. $1 \leq n \leq 10^5$
2. $1 \leq K \leq n$

Wyjście

W pierwszym i jedynym wierszu program powinien wypisać jedną liczbę całkowitą - najmniejszą możliwą liczbę kroków, po wykonaniu których wszystkie klocki będą odwrócone białą stroną, lub -1 jeżeli jest to niemożliwe.

Przykład

Dla danych wejściowych:

8 3
bbbwbwb

poprawną odpowiedzią jest:

3

Kolejne kroki:

odwracamy klocki o numerach 0, 1, 2: wynik wwwbwwb
odwracamy klocki o numerach 4, 5, 6: wynik wwwwbwb
odwracamy klocki o numerach 5, 6, 7: wynik wwwwww

3. Uncrossed lines (25 punktów)

Zadanie

Dane są dwie tablice liczb całkowitych T_1 i T_2 o długościach odpowiednio n_1 i n_2 . Zapisujemy liczby zawarte w obu tablicach w dwóch oddzielnych liniach (zachowując ich oryginalną kolejność).

Możemy łączyć obie linie rysując odcinki między liczbami $T_1[i]$ i $T_2[j]$ jeżeli:

1. $T_1[i] = T_2[j]$.
2. Narysowane odcinki nie przecinają się (nawet w punktach końcowych, czyli dana liczba może należeć tylko do jednego odcinka).

Napisz program, który znajduje i wypisuje największą możliwą liczbę odcinków łączących jakie możemy narysować z zachowaniem powyższych warunków.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite, n_1 i n_2 - długości odpowiednio tablic T_1 i T_2 . W kolejnych dwóch liniach znajduje się odpowiednio n_1 i n_2 liczb całkowitych - wartości tablic T_1 i T_2 .

Ograniczenia

1. $1 \leq n_1, n_2 \leq 500$
2. $1 \leq T_1[i], T_2[j] \leq 2000, i = 0, \dots, n_1 - 1, j = 0, \dots, n_2 - 1$

Wyjście

W pierwszym i jedynym wierszu program powinien wypisać jedną liczbę całkowitą - największą możliwą liczbę odcinków łączących jakie możemy narysować.

Przykład

Dla danych wejściowych:

```
6 5
1 3 7 1 7 5
1 9 2 5 1
```

poprawną odpowiedzią jest:

2

W tym przypadku możemy połączyć (na przykład) elementy o indeksach 0, 0 (dwie jedynki) oraz 5, 3 (dwie piątki).

4. Maximal dot product (25 punktów)

Zadanie

Dane są dwie tablice liczb całkowitych, `nums1` i `nums2`.

Napisz program, który wyznacza i wypisuje maksymalny iloczyn skalarny dwóch niepustych podciągów tablic `nums1` i `nums2`. Ze względu na definicję iloczynu skalarnego wybieramy zawsze podciągi o tej samej długości.

Podciąg tablicy to nowa tablica, która jest utworzona z tablicy oryginalnej przez usunięcie części (lub żadnego) elementów bez zmiany kolejności pozostałych elementów (czyli `[2, 3, 5]` jest podciągiem tablicy `[1, 2, 3, 4, 5]`, ale `[1, 5, 3]` nie jest).

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite, n_1 (liczba elementów tablicy `nums1`) i n_2 (liczba elementów tablicy `nums2`). W kolejnych dwóch liniach znajduje się odpowiednio n_1 i n_2 liczb całkowitych - wartości tablic `nums1` i `nums2`.

Ograniczenia

- $1 \leq n_1, n_2 \leq 500$
- $-1000 \leq \text{nums1}[i], \text{nums2}[i] \leq 1000$

Wyjście

W pierwszym i jedynym wierszu program powinien wypisać jedną liczbę całkowitą - największą możliwą wartość iloczynu skalarnego podciągów tablic `nums1` i `nums2`.

Przykład 1

Dla danych wejściowych:

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

poprawną odpowiedzią jest:

18

1. Bierzemy podciąg o wartościach 2 i -2 z tablicy `nums1`
2. Bierzemy podciąg o wartościach 3 i -6 z tablicy `nums2`
3. Iloczyn skalarny tych podciągów to $2 \times 3 + (-2) \times (-6) = 18$

Przykład 2

Dla danych wejściowych:

```
2 2
-1 -1
1 1
```

poprawną odpowiedzią jest:

-1

1. Bierzemy podciąg o wartości -1 z tablicy `nums1`
2. Bierzemy podciąg o wartości 1 z tablicy `nums2`
3. Iloczyn skalarny tych podciągów to $(-1) \times 1 = -1$