

1. Sekwencje binarne

Zadanie

Rozważamy wszystkie sekwencje o długości N , składające się tylko z elementów 0 i 1, w których dwie jedynki nie mogą ze sobą sąsiadować (czyli 110 nie jest poprawną sekwencją o długości 3, a 0101 jest poprawną sekwencją o długości 4).

Napisz program, który znajduje K -tą sekwencję w alfabetycznie posortowanym zbiorze sekwencji o długości N .

Wejście

Na wejściu znajdują się dwie dodatnie liczby całkowite, $1 \leq N \leq 63$ i $1 \leq K \leq 100000$.

Wyjście

Program powinien wypisać znalezioną sekwencję lub -1, jeżeli K jest większe od liczby poprawnych sekwencji.

Przykład:

Dla danych

3 1

Program powinien wypisać:

000

2. Odcinek

Zadanie

Dana jest tablica kwadratowa T o boku n zawierająca wartości całkowite. Szukamy w tablicy T takiego ciągu k sąsiednich elementów (położonych w wierszu, kolumnie lub na przekątnej - prawo lub lewoskośnie), których suma wartości T_{ij} jest największa. Stosujemy zasadę periodycznych warunków brzegowych: każdy element tablicy ma dokładnie 8 najbliższych sąsiadów. Na przykład sąsiadami elementu T_{00} są elementy T_{10} , T_{11} , T_{01} , $T_{(n-1)1}$, $T_{(n-1)0}$, $T_{(n-1)(n-1)}$, $T_{0(n-1)}$, $T_{1(n-1)}$. W związku z tym odcinek może leżeć częściowo poza tablicą: w tym przypadku tablica jest powielana w odpowiednim kierunku (kierunkach).

Napisz program, który:

1. Wczyta rozmiar tablicy, n , długość odcinka, k , i tablicę T ,
2. Wyznaczy optymalne położenie odcinka,
3. Wypisze maksymalną sumę k sąsiednich elementów T .

Wejście:

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite $1 \leq n \leq 100$ i $2 \leq k \leq n$. Kolejne n wierszy zawiera po n liczb całkowitych (wiersze T).

Wyjście:

Program powinien wypisać jedną liczbę całkowitą - największą sumę k sąsiednich wartości elementów tablicy T .

Przykład:

Dla danych wejściowych:

```
5 2
8 1 0 1 0
2 3 4 7 1
```

2 7 4 0 1
1 8 5 1 4
0 1 6 3 9

poprawną odpowiedzią jest:

17

Maksymalny odcinek tworzą elementy T_{00} i T_{44} .

3. Stamp

Zadanie

Dana jest tablica kwadratowa T o boku n zawierająca wartości całkowite oraz tablica prostokątna P o wymiarach $k \times l$ przechowująca wartości 0 lub 1. Tablice P możemy “nałożyć” na tablicę T tak, by przykryła ona pewien jej fragment (ale musi mieścić się całkowicie w obrębie T). Szukamy takiego położenia tablicy P , że suma elementów T_{ij} przykrytych przez elementy P o wartości 1 była największa.

Napisz program, który:

1. Wczyta rozmiar tablicy T , n , rozmiary tablicy P , k i l a następnie tablice T i P ,
2. Wyznaczy optymalne położenie P ,
3. Wypisze maksymalną sumę elementów T przykrytych przez elementy P równe 1.

Wejście:

W pierwszym wierszu standardowego wejścia znajdują się trzy liczby całkowite $1 \leq n \leq 100$ i $1 \leq k, l < n$. Kolejne n wierszy zawiera po n liczb całkowitych (wiersze T). Następne k wierszy (po l elementów 0 lub 1) stanowią wiersze tablicy P .

Wyjście:

Program powinien wypisać jedną liczbę całkowitą - największą sumę “przykrytych” wartości tablicy T .

Przykład:

Dla danych wejściowych:

```
5 2 2
5 1 0 1 0
2 3 4 7 1
2 7 4 0 1
1 8 5 1 9
0 1 6 3 0
0 1
1 0
```

poprawną odpowiedzią jest:

12