

TASK 2.1. RACE

Primarul oraşului Plovdiv a decis să organizeze o cursă pe străzile oraşului pentru a demonstra că acestea sunt potrivite pentru viteze mari. El trebuie să aleagă o rută pentru concurs care să fie cât mai rapidă cu putinţă. După ce a vorbit cu consilierii săi el a stabilit următoarele restricţii. Cursa trebuie să înceapă şi să se termine în intersecţia unde se află primăria. Singurele schimbări de direcţie acceptate de-a lungul traseului sunt cele spre stânga (a face o schimbare de direcţie spre stânga faţă de direcţia curentă înseamnă să schimbăm direcţia cu un unghi mai mare sau egal cu 0, dar strict mai mic ca 180 de grade). Dacă există mai multe rute ce îndeplinesc condiţiile de mai sus ruta aleasă va fi aceea care are cea mai scurtă stradă cât mai lungă posibil.

Oraşul Plovdiv are N intersecţii şi M străzi bidirecţionale ce le leagă. Intersecţiile sunt descrise prin două coordonate în plan şi sunt numeroase de la 1 la N în ordinea în care sunt date la intrare. Primăria se află situată în intersecţia cu numărul 1. Străzile sunt linii drepte ce încep într-o intersecţie şi se termină în alta. Lungimea unei străzi este egală cu distanţa euclidiană dintre capetele ei. O stradă este definită prin numerele capetelor ei. Nu există mai mult de o stradă între două intersecţii. Străzile se intersectează doar în capetele lor.

Scrieţi un program numit **RACE** care găseşte o rută, din cele posibile în oraş, care începe şi se termină cu intersecţia 1 şi din fiecare intersecţie se merge fie înainte, fie se schimbă direcţia spre stânga, iar cea mai scurtă stradă din această rută este cât mai lungă cu putinţă. Ruta nu poate trece de două ori în acelaşi sens pe aceeaşi stradă.

Prima linie a **dispozitivului standard de intrare** conţine două numere $N(3 \leq N \leq 2000)$ şi $M(5 \leq M \leq 25000)$ separate printr-un spaţiu. Fiecare din următoarele N linii conţine coordonatele X şi Y ale intersecţiei date, separate printr-un spaţiu. Coordonatele sunt numere întregi din intervalul $[-10000, 10000]$. Ultimele M linii descriu străzile. Fiecare dintre aceste linii conţine două numere separate printr-un spaţiu, acestea reprezentând numerele de ordine a intersecţiilor ce mărginesc strada.

La **dispozitivul standard de ieşire** se va scrie descrierea rutei. Prima linie va conţine numărul de intersecţii din rută (intersecţia 1 fiind numărată de două ori). Următoarea linie va conţine numerele intersecţiilor de-a lungul rutei (începând şi terminând cu intersecţia 1), în ordinea corespunzătoare parcurgerii rutei, separate prin spaţiu.

Există cel puţin o soluţie a problemei. Dacă există mai multe soluţii tipăriţi oricare dintre ele.

EXAMPLU

Input

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

Output

```
5
1 2 5 4 1
```

TASK 2.2. GAME

Micului Ion îi place să joace în timpul liber. Din nefericire, el nu se poate bucura întotdeauna de compania prietenilor și câteodată se plictisește fiind singur. Astfel, Ion inventează jocuri, în care este prezent un singur jucător. El este mândru de ultimul său joc și ar vrea să vă povestească și vouă despre el.

Se dau două șiruri finite de numere întregi pozitive. Jocul constă din mișcări consecutive. Este permisă următoarea mișcare. Se șterg ultimele K_1 numere ($K_1 \geq 1$) din primul șir (posibil tot șirul) calculându-se suma lor S_1 , și ultimele K_2 numere ($K_2 \geq 1$) din al doilea șir (posibil tot șirul) calculându-se suma lor S_2 . Apoi se determină costul mișcării ca fiind $(S_1 - K_1) * (S_2 - K_2)$. Mișcările se repetă până la eliminate tuturor numerelor din ambele șiruri. Costul total al jocului este suma costurilor tuturor mișcărilor. Scopul tău este să minimizezi acest cost total. Nu se permite să lași unul din șiruri vid, în timp ce celălalt mai conține elemente.

Fiind cunoscute regulile jocului, ai înțeles că problema este ușor rezolvabilă cu ajutorul calculatorului, așa că urmează să scrii programul **GAME**, care calculează cel mai mic cost total al jocului.

Datele de intrare sunt citite de la **dispozitivul standard de intrare** și formează trei linii. Prima linie conține două numere întregi separate prin spațiu, L_1 și L_2 ($1 \leq L_1, L_2 \leq 2000$), care reprezintă lungimile celor două șiruri. Al doilea rând conține L_1 numere întregi separate prin spațiu - elementele primului șir. A treia linie conține L_2 numere întregi separate prin spațiu - elementele celui de-al doilea șir. Valorile elementelor din ambele șiruri nu vor depăși 1000.

Programul va afișa la **dispozitivul standard de ieșire** o linie care va conține un singur număr – costul minim total al jocului descris mai sus.

Exemplu

Intrare	Ieșire
3 2	2
1 2 3	
1 2	

TASK 2.3. CODES

Vom defini distanța $d_H(X, Y)$ dintre două șiruri X și Y de lungimi egale ca fiind distanța Hamming, adică numărul de poziții prin care X și Y diferă.

De exemplu, $d_H(1001, 0010)=3$ și $d_H(1001111, 0010101)=4$.

Fie $n \geq 1$. Presupunem că $C = W_1, W_2, \dots, W_M$ este o listă de M șiruri binare de lungime n .

Considerăm că C este o listă circulară și vom defini distanța $d_C(W_i, W_j)$ dintre două șiruri W_i și W_j din listă ca fiind $d_C(W_i, W_j) = \min\{\text{abs}(i-j), M-\text{abs}(i-j)\}$.

Fie k satisfăcând relația $1 \leq k < n$. Vom spune că C este un **cod circular de lungime n și anvergură k** dacă pentru oricare i, j , $1 \leq i, j \leq M$ sunt adevărate relațiile:

- (1) Dacă $d_C(W_i, W_j) \leq k$, atunci $d_H(W_i, W_j) = d_C(W_i, W_j)$;
- (2) Dacă $d_C(W_i, W_j) > k$, atunci $d_H(W_i, W_j) > k$.

Problema principală în studiul codurilor circulare este determinarea numărului maxim de șiruri dintr-un cod circular de lungime n și anvergură k . Valoarea exactă a acestui număr este cunoscută doar pentru câteva valori mici ale parametrilor n și k . Urmează să construiți un cod circular conținând cât mai multe șiruri posibile, pentru perechi de valori date ale parametrilor n, k .

Test #	1	2	3	4	5	6	7	8	9	10
n	5	6	6	7	7	8	8	9	10	10
k	1	1	2	1	2	1	2	2	2	3

Trebuie să trimiteți 10 fișiere conținând codurile construite utilizând perechile de parametri din tabela de mai sus.

Nu trebuie să trimiteți nici un program!

Prima linie a fișierului trimis va conține

#FILE code t

unde t este numărul testului.

Următoarele M linii vor conține șiruri consecutive ale codului construit de lungime n și anvergură k .

Pentru fiecare test, cea mai bună dintre toate soluțiile concurenților va primi 10 puncte. Dacă cea mai bună soluție este un cod din B șiruri și ați trimis o soluție corectă cu M șiruri, punctajul va fi de $10M/B$. Punctajul pentru fiecare test va fi rotunjit la prima cifră după virgulă. Punctajul total va fi rounjit la cel mai apropiat întreg.

EXEMPLU

$n=4, k=1, M=8$

```
#FILE code 0
1101
1100
1110
1010
1011
0011
0001
0101
```