

## E. Chińska komórka

---

Dostępna pamięć: 32 MB

Niezbyt dawno temu wpisywanie tekstu na telefonie komórkowym wyglądało następująco.  $L$  liter napisanych w kolejności alfabetycznej było podzielone pomiędzy  $K$  klawiszy, tj. każdy klawisz zawierał spójny fragment alfabetu. Aby wpisać określoną literę, należało znaleźć klawisz zadaną literą; jeśli stała ona na nim na  $i$ -tej pozycji, należało nacisnąć ten klawisz  $i$  razy. Przykładowo na standardowej komórce  $L = 26$  liter było podzielonych między  $K = 8$  klawiszy. Na klawiszu „7” znajdowały się litery **pqrs**. Wprowadzenie litery **r** wymagało zatem naciśnięcia tego klawisza 3 razy.

Ostatnio Rząd Chińskiej Republiki Ludowej postanowił wprowadzić na rynek podobną komórkę, tylko nieco większą, żeby zmieściły się na niej wszystkie chińskie symbole (dalej nazywane literami). Litery zostały ponumerowane od 1 do  $L$ , pozostało je tylko podzielić pomiędzy  $K$  klawiszy. Rząd ChRL ustalił, że układ klawiszy na komórce ma zostać tak zoptymalizowany, żeby pewien określony komunistyczny pamflet dało się wpisać naciskając jak najmniejszą liczbę klawiszy.

### Specyfikacja danych wejściowych

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne  $K$  i  $L$  oddzielone pojedynczą spacją, spełniające warunki:  $1 \leq K \leq 100$ ,  $1 \leq L \leq 10\,000$ ,  $K \leq L$ . W drugim wierszu znajduje się  $L$  liczb naturalnych  $f_1, f_2, \dots, f_L$  oddzielonych pojedynczym odstępem. Liczba  $f_i \in [1, \dots, 1000]$  jest liczbą wystąpień litery  $i$  w pamflecie. W siedmiu punktowanych testach zachodzi dodatkowo  $L \leq 1000$ .

### Specyfikacja danych wyjściowych

W pierwszym wierszu wyjścia Twój program powinien wypisać jedną liczbę naturalną będącą minimalną liczbą naciśnieć klawiszy konieczną do wpisania pamfletu na najlepszej możliwej klawiaturze składającej się z  $K$  klawiszy. W drugim wierszu wyjścia Twój program powinien wypisać opis takiej klawiatury:  $K$  liczb naturalnych  $k_1, k_2, \dots, k_K$  oddzielonych pojedynczymi odstępami, gdzie  $k_j$  jest liczbą liter przypisanych do klawisza  $j$ .

Jeśli istnieje wiele optymalnych rozmieszczeń liter na klawiszach, należy wybrać takie, które maksymalizuje liczbę liter na ostatnim klawiszu, wśród nich to, które maksymalizuje liczbę liter na przedostatnim klawiszu itd.

### Przykład A

Wejście:

5 6  
2 2 2 2 2 2

Wyjście:

14  
1 1 1 1 2

### Przykład B

Wejście:

2 4  
3 1 1 1

Wyjście:

8  
2 2

### Przykład C

Wejście:

3 6  
3 4 3 1 1 4

Wyjście:

23  
2 3 1