

1. Sudoku (20 punktów)

Zadanie

Sudoku składa się kwadratu o wymiarach 9×9 podzielonego na 9 małych kwadratów o wymiarach 3×3 . Na potrzeby zadania, kolejne małe kwadraty numerujemy wierszami od 1 do 9. W poprawnym rozwiązaniu: w każdym wierszu, każdej kolumnie i każdym małym kwadracie znajdują się cyfry 1 – 9. W poprawnym rozwiązaniu ktoś zamienił miejscami dwa małe kwadraty. Proszę napisać program, który wskaże zamienione kwadraty.

Wejście

W 9 wierszach umieszczono standardowego wejścia umieszczono kolejne wiersze sudoku.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać dwa numery małych kwadratów, które zostały zamienione. Numery te należy wypisać w porządku rosnącym.

Przykłady

Dla danych wejściowych:

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

Poprawną odpowiedzią jest

5 9

Zostały zamienione miejscami kwadrat środkowy (5) z prawym dolnym kwadratem (9)

2. Przekątne (20 punktów)

Zadanie

Dany jest ciąg $1, 1, 2, 3, 5, 8, 13, 21, 24, 25, 29, 34, 43, 47, 50, \dots$, którego pierwsze dwa wyrazy są równe jeden, a każdy kolejny jest określony wzorem: $A_n = A_{n-2} \bmod 10 + A_{n-1}$. Dana jest tablica kwadratowa o rozmiarze $N \times N$. Tablicę uzupełniamy kolejnymi wyrazami ciągu po spirali zgodnej z ruchem wskazówek zegara rozpoczynając od lewego górnego rogu. Proszę napisać program, który dla tak wypełnionej tablicy zwraca sumę elementów leżących na obu przekątnych.

Wejście

Pierwszy wiersz zawiera liczbę $3 \leq N \leq 40$, będącą rozmiarem tablicy. Drugi wiersz zawiera numer $1 \leq P \leq 10$ wyrazu ciągu od którego rozpoczynamy wpisywanie kolejnych wyrazów do tablicy.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać wartość sumy elementów leżących na przekątnych tablicy.

Przykłady

Dla danych wejściowych:

3
7

Poprawną odpowiedzią jest

159

Wypełniona tablica wygląda następująco:

13 21 24
47 50 25
43 34 29

3. Przedziały (20 punktów)

Zadanie

Dany jest ciąg przedziałów domkniętych $[a_1, b_1], \dots, [a_n, b_n]$. Początki i końce przedziałów są nieujemnymi liczbami całkowitymi. Proszę znaleźć przedział, który zawiera w sobie największą liczbę innych przedziałów. Dane do zadania są generowane przez funkcję $\text{gen}(X, Y)$.

```
void gen(int X, int Y) {
    int p = 6;
    for (int i=0; i<X; i++) {
        do {
            p = (p*134775813+1) % 4294967296;
            a = p % Y;
            p = (p*134775813+1) % 4294967296;
            b = p % Y;
        } while (b>=a);
        cout << a << " " << b << endl;
    }
}
```

Wejście

Pierwszy i jedyny wiersz zawiera liczby X, Y będące parametrami funkcji generującej przedziały. Liczby X, Y są z zakresu $[5..10^6]$.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać liczbę przedziałów zawierającą się w znalezionym przedziale.

Przykłady

Dla danych wejściowych:

5 10

Poprawną odpowiedzią jest

3

Funkcja $\text{gen}(5, 10)$ generuje 5 przedziałów: $[3, 6], [3, 6], [5, 6], [3, 4], [1, 4]$
Pierwszy z przedziałów zawiera przedziały: 2, 3 i 4.

4. Rurociąg (20 punktów)

Zadanie

Dany jest zestaw N rur o długościach wyrażonych całkowitą liczbą metrów L_1, L_2, \dots, L_N . Spośród tego zestawu chcemy zbudować rurowciąg, które długość wyniesie S metrów. Proszę napisać program, który wyznacza minimalną liczbę rur niezbędną do zbudowania rurowciągu.

Wejście

Pierwszy wiersz zawiera liczbę $3 \leq N \leq 40$, będącą liczbą rur. Drugi wiersz zawiera kolejno liczby: L_1, L_2, \dots, L_N . Liczby L_i są z zakresu $[1..50]$. Trzeci wiersz zawiera liczbę S będącą długością rurowciągu.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać minimalną liczbę rur niezbędną do budowy rurowciągu. Jeżeli rurowciągu nie da się zbudować powinien wypisać słowo BRAK.

Przykłady

Dla danych wejściowych:

```
5
2 7 4 3 11
16
```

Poprawną odpowiedzią jest

```
3
```

Są to rury o długościach: 2, 3, 11

5. Przelot (20 punktów)

Zadanie

Układ planetarny Algon składa się z N planet o numerach od 0 do $N - 1$. Niestety własności fizyczne układu powodują, że nie da się łatwo przelecieć między dowolnymi dwiema planetami. Na szczęście mozolna eksploracja kosmosu doprowadziła do stworzenia listy E dopuszczalnych bezpośrednich przelotów. Każdą taką trasę opisuje trójka postaci (u, v, t) , gdzie u i v to numery planet (można założyć, że $u < v$) a t to czas podróży między nimi (przelot z u do v trwa tyle samo co z v do u). Dodatkową nietypową własnością układu Algon jest to, że niektóre planety znajdują się w okolicy osobliwości. Znajdując się przy takiej planecie możliwe jest zagięcie czasoprzestrzeni umożliwiające przedostanie się do dowolnej innej planety leżącej przy osobliwości w czasie zerowym.

Należy napisać program który znajdzie najkrótszy czas podróży z planety A do planety B , mając do dyspozycji listę możliwych bezpośrednich przelotów oraz listę planet znajdujących się koło osobliwości.

Wejście

W pierwszym wierszu znajdują się kolejno: liczba planet N , liczba tras E , liczba osobliwości S , planeta startowa A i planeta docelowa B . Kolejne E wierszy zawiera parametry tras w postaci trójek (u,v,t) . Następne S wierszy to numery planet z osobliwościami.

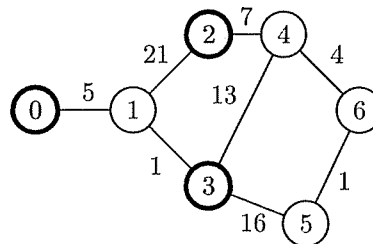
Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać długość minimalnej drogi z punktu A do B . Jeżeli droga taka nie istnieje program powinien wypisać słowo BRAK.

Przykłady

Dla danych wejściowych:

```
7 8 3 1 5
0 1 5
1 2 21
1 3 1
2 4 7
3 4 13
3 5 16
4 6 4
5 6 1
0
2
3
```



Poprawną odpowiedzią jest liczba 13. Odwiedzamy po kolei planety 1, 3, 2, 4, 6 i kończymy na planecie 5 (z planety 2 do 3 dostajemy się przez zagięcie czasoprzestrzeni).