

Pole wielokąta

Liczba punktów: 60

Limit czasu: 1-3s

Limit pamięci: 256MB

Oblicz pole wielokąta wypukłego.

Wielokąt wypukły jest to wielokąt, który dla dowolnych jego dwóch punktów zawiera również odcinek łączący te punkty.

Podpowiedź:

Punkty są poukładane przeciwnie do ruchu wskazówek zegara, punkty mają współrzędne całkowite.

Wejście

Pierwsza linia zawiera liczbę kolejnych linii. W kolejnych liniach, pierwsza liczba w linii zawiera liczbę punktów wielokąta. Następnie po spacjach wypisane są współrzędna x-owa pierwszego punktu, współrzędna y-owa pierwszego punktu, współrzędna x-owa drugiego punktu, współrzędna y-owa drugiego punktu, itd.

Wyjście

Podłoga z pola wielokąta.

Przykład

Wejście

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

Wyjście

```
5
2
11
3
5
```

Operacje na zbiorach

Liczba punktów: 80

Limit czasu: 1-3 s

Limit pamięci: 256 MB

Zbiór jest podstawowym pojęciem w matematyce. Zbiór, w uproszczeniu, to zestaw elementów. W zbiorze nie mogą istnieć dwa takie same elementy. Jeśli element jest w zbiorze, to mówimy że należy do zbioru. Na zbiorach można wykonywać podstawowe operacje:

- suma zbiorów A i B - to zbiór złożony z tych i tylko tych elementów, które należą do zbioru A lub do zbioru B
- iloczyn zbiorów A i B - to zbiór złożony z tych i tylko tych elementów, które należą do zbioru A i do zbioru B
- różnica zbiorów A i B - to zbiór złożony z tych i tylko tych elementów, które należą do zbioru A i nie należą do zbioru B

Należy stworzyć prosty kalkulator wyznaczający zbiory.

Początkowym zbiorem będzie podzbiór kolejnych liczb naturalnych od 1 do n . Kalkulator będzie zawierał zbiory jako

- zmienne (ciąg znaków łańciskich, w których wielkość ma znaczenie)
- napisy : {ciąg liczb przedzielony białymi znakami}
- pojedyncze liczby

Zbiory można zmieniać operatorami sumy (+), różnicy (-), iloczynu (*). Wszystkie operatory są lewostronnie domknięte tj. napis $a=b-c-d$ jest równoznaczny z $a=(b-c)-d$. Najwyższy priorytet ma iloczyn, natomiast suma i różnica mają takie same priorytety. Kalkulator zawiera także nawiasy i operator przypisania. Specjalnym znacznikiem jest %, który oznacza zbiór złożony, z wszystkich liczb od 1 do n . Jeśli zbiór nie był wcześniej przypisany, to traktujemy go jak zbiór pusty (czyli zbiór bez elementów).

Wejście

W pierwszej linii znajduje się liczba k oraz k zmiennych. W drugiej linii znajdują się dwie liczby n i s . Liczba n to wyżej opisane ograniczenie na liczbę elementów zbioru ($n < 1000$). W następnych s liniach będą znajdowały się opisy operacji. Każda z s linii zawiera zmienną, znak = oraz ciąg operacji podobny do zapisu w języku C.

Wyjście

Wyjście składa się z k linii opisujących wartościowanie k zmiennych opisanych w pierwszej linii danych wejściowych. Jeśli zmienna była przypisana, wtedy wpisujemy zbiór, przy czym wartości ze zbioru wypisujemy posortowane rosnąco. Jeśli zmienna nie była przypisana, wtedy wpisujemy w linii 'brak'.

Przykład

Wejście

7 a A b bbb d B C

6 7

a = % - d

A = {1 4 2} +{1 6 }

B = A + {3}

C =A + 3

B = B*B+B

e = B * (C+{5})

bbb = d - e

Wyjście

{ 1 2 3 4 5 6 }

{ 1 2 4 6 }

brak

{ }

brak

{ 1 2 3 4 6 }

{ 1 2 3 4 6 }

Rosnący podciąg

Punkty: 80

Limit czasu: 1-3s

Limit pamięci: 256MB

Napisz program szukający najdłuższego niemalejącego podciągu (podciąg może zawierać "przerwy") w danym ciągu liczb.

Innymi słowy, z danego ciągu staramy się wykreślić jak najmniej liczb tak, aby pozostałe liczby tworzyły sekwencję niemalejącą.

Wejście

Na wejściu podana zostanie liczba elementów ciągu $1 \leq n \leq 1000000$ oraz n liczb a_i ($0 \leq a_i \leq 2^{31}-1$) dla $i=1, \dots, n$, będących kolejnymi elementami ciągu. Wszystkie liczby rozdzielone będą jednym lub dwoma białymi znakami (spacja, nowa linia, powrót karetki).

Wyjście

Na wyjściu należy wypisać jedną liczbę, będącą długością najdłuższego niemalejącego podciągu.

Przykład

Wejście

```
30
1 4 9
2 5 10
3 6 11
4 7 12
18 28 38
17 27 37
16 26 36
15 25 35
14 24 34
13 23 33
```

Wyjście

```
9
```

Wejście

```
9
1 5 10 15 20 25
2 3 4 5
```

Wyjście

```
6
```

W pierwszym przypadku możemy wybrać podciąg 1 2 3 4 7 12 13 23 33.

W drugim przykładzie wybranym ciągiem jest 1 5 10 15 20 25.

Gra króla Artura

Liczba punktów: 90

Limit czasu: 1-3 s

Limit pamięci: 256 MB

Król Artur i rycerze okrągłego stołu lubią pograć w karty.

Gra w którą grają jest bardzo skomplikowana, ale najważniejsze jest to, że osoba, która wygrywa, otrzymuje punkty wraz z wszystkimi osobami, które siedzą obok niej w pewnym sąsiedztwie.

Rycerze grają kilka rund i na koniec chcą wiedzieć ile zdobyli punktów. Jako, że nie mają głowy do liczb, to na Ciebie spada obowiązek zliczenia punktów każdego z graczy.

Dla ułatwienia rycerze są ponumerowani od 1 do n i osoby z sąsiednimi numerami siedzą obok siebie. Oczywiście 1 siedzi obok 2 i n .

Wejście

W pierwszej linii wystąpią dwie liczby n i r , gdzie n to liczba osób grających (nie większa niż 100000), a r to liczba rund (nie większa niż 50000).

W kolejnych r liniach znajdują się trzy liczby c w d , gdzie c to numer rycerza, który wygrał; d to liczba sąsiadów po obu stronach, którzy otrzymują punkty; w to liczba punktów które otrzymują.

Wyjście

Należy wypisać n liczb - liczby punktów, które zdobyli gracze w kolejności 1 gracz, 2 gracz itd. Suma punktów każdego z graczy nie przekroczy zakresu int.

Przykład

Wejście

```
10 4
1 1 2
6 2 7
4 8 0
10 3 1
```

Wyjście

```
6 3 3 10 2 2 2 2 6 6
```

Dominacja

Punkty: 100

Limit czasu: 1-2s

Limit pamięci: 256MB

Należy znaleźć rozmiar najmniejszego zbioru dominującego w drzewie. Podzbiór wierzchołków w grafie jest dominujący, jeżeli każdy wierzchołek grafu albo należy do zbioru dominującego, albo ma sąsiada w tym zbiorze.

Graf jest drzewem jeżeli jest spójny i nie zawiera żadnego cyklu.

Wejście

Na wejściu podana zostanie liczba wierzchołków grafu $1 \leq n \leq 1000000$. Następnie pojawi się $n - 1$ par liczb

$u_i \ v_i$

oznaczających, że u_i oraz v_i są połączone krawędzią; $0 \leq u_i, v_i < n$.

Wyjście

Na wyjściu należy wypisać jedną liczbę, będącą rozmiarem najmniejszego zbioru dominującego w drzewie.

Przykład

Wejście

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

Wyjście

```
2
```

Wejście

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

Wyjście

```
1
```

W pierwszym przypadku możemy wybrać drugi i czwarty wierzchołek na ścieżce.

W drugim przykładzie wybranym zbiorem dominującym jest wierzchołek 0, który sąsiaduje z każdym innym.