VI Warmińsko-Mazurskie Zespołowe Zawody Programistyczne - Wydział Matematyki i Informatyki, UWM

19 lutego 2015

Instrukcja techniczna

Rozwiązania zadań, oceniane są automatycznie przez serwer SPOJ. Rozwiązania (w postaci kodu źródłowego), należy wysyłać po uprzednim zalogowaniu się na stronie serwera. Adres konkursu: www.spoj.com/VIWMZZP

Wszystkie zadania, są punktowane jednakowo. O zajętym miejscu przez drużynę decyduje przede wszystkim, liczba poprawnie rozwiązanych zadań. W przypadku gdy drużyny posiadają taką samą liczbę punktów, czynnikiem decydującym, jest łączny czas, liczony od momentu rozpoczęcia zawodów do chwili oddania ostatniego poprawnie rozwiązanego zadania. Za oddanie niepoprawnie rozwiązanego zadania, drużyna otrzymuje karę 20 minut, doliczoną do jej łącznego czasu.

Każde zadanie, posiada ograniczenie czasowe. Aby zadanie zostało uznane za zaliczone, oddany do oceny program, musi wykonać się w czasie krótszym niż 10 sekund. Ograniczenie, co do użycia pamięci operacyjnej wynosi 256 MB. Czas wykonania programu, tak jak i zużycie pamięci, nie wpływają na liczbę otrzymanych punktów. Liczy się przede wszystkim poprawne rozwiązanie.

Wszystkie operacje wejścia/wyjścia, należy dokonywać, z wykorzystaniem standardowych urządzeń wejścia/wyjścia, tzn. czytanie danych i wypisywanie danych, musi odbywać się z użyciem konsoli. Zabrania się dokonywania operacji na plikach.

W zadaniach, odpowiedzi powinny być podane w osobnych liniach. Każda odpowiedź, musi zaczynać się od początku wiersza (bez spacji) i kończyć znakiem nowej linii. Należy unikać wstawiania białych znaków do odpowiedzi. Dokonać tego można tylko w przypadku, gdy treść zadania tego wymaga.

Poniżej znajdują się przykłady prawidłowego wysłania napisu "dane" na urządzenie wyjścia.

$C\backslash C++$
printf("dane \n");
std::cout«"dane"«std::endl;

cout«"dane"«endl;

Pascal
writeln('dane');

Po wysłaniu programu na serwer SPOJ, po kilku chwilach ukaże się jeden z poniższych rezultatów:
• AC - zadanie wykonane poprawnie,
• WA - program wykonał się poprawnie, ale zwrócił błędną odpowiedź,
• TLE - program nie zakończył działania w wyznaczonym czasie,
• CE - błąd kompilacji,
• RE - błąd w trakcie wykonania programu:
⟼ SIGSEGV - "segmentation fault", np. indeks poza tablicą,
⟼ SIGXFSZ - program wygenerował za dużą ilość danych,
⟼ SIGFPE - "floating point error", np. dzielenie przez zero,
⟼ SIGABRT - błąd zgłaszany przez program zdarza się np. w C++ STL,
⟼ NZEC - non-zero exit code, program nie zwrócił zera po wykonaniu,
⟼ inne - pozostałe błędy.

Zadanie1 - Obwody trójkątów

Dane są położenia kilku figur. Każda figura, jest opisana przez trzy różne punkty, będące jej wierzchołkami. Należy obliczyć obwody podanych figur. Odległość między punktami, obliczamy za pomocą metryki euklidesowej. Przykładowo dla punktów $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$, odległość euklidesową określa się jako

$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Wejście: W pierwszym wierszu znajduje się liczba figur. W kolejnych wierszach znajdują sie położenia wierzchołków figur. W pojedynczym wierszu, znajdują się dane dotyczące pojedynczej figury.

Wyjście: Każdy wiersz zawiera obwód figury zaokrąglony do dwóch miejsc po przecinku.

Przykładowe dane wejściowe	Wynik do przykładu
4	30.97
−5 4 5 2 −1 −6	27.63
−5 4 5 2 −1 −4	23.31
−5 4 5 2 −1 −1	26.2
−6 4 5 2 −1 −2	

Zadanie2 - Snap

Ciąg n elementowy ($n > 1$), liczb całkowitych, nazwiemy „snapem", jeżeli bezwzględne wartości różnic między kolejnymi elementami, przyjmują wszystkie możliwe wartości pomiędzy 1 a $n - 1$.

Przykład:

3 7 5 2 3

Powyższy ciąg nazwiemy snapem, gdyż bezwzględne różnice między elementami są kolejno 4, 2, 3, 1, czyli wszystkie wartości pomiędzy 1 a 4 $(5-1)$. Napisz program określający dla każdego z pewnej liczby ciągów, czy jest on snapem (odpowiedź: Tak), czy też nie (odpowiedź: Nie).

Wejście: Pierwszy wiersz zawiera liczbę ciągów do sprawdzenia. Kolejne wiersze są w formacie: zaczynają się od pewnej liczby $n \leqslant 3000$, po której następuje n liczb całkowitych stanowiących kolejne wyrazy ciągu.

Wyjście: Dla każdego wiersza danych wejściowych wygeneruj wiersz o treści "Tak" lub "Nie".

Przykładowe dane wejściowe	Wynik do przykładu
2 4 1 4 2 3 5 1 4 2 −1 6	*Tak* *Nie*

Zadanie3 - Problem $3+n$

Rozważmy następujący algorytm generujący ciąg liczb całkowitych. Zaczyna się od liczby n. Jeżeli n jest parzyste, dzielimy je przez 2. W innym przypadku mnożymy przez 3 i dodajemy 1. Obliczenia powtarzamy dla nowej wartości n. Kończymy, gdy $n = 1$. Na przykład dla $n = 22$ generowany jest następujący ciąg liczb:

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Przypuszcza się (choć nie zostało to jeszcze udowodnione), że algorytm ten zatrzymuje się z $n = 1$ dla dowolnej liczby początkowej n. Rzeczywiście, hipoteza ta sprawdza się dla wszystkich liczb do co najmniej 1 000 000.

Dla danej liczby n długość wygenerowanego ciągu, włączając końcową liczbę 1, nazywamy długością cyklu dla n. W powyższym przykładzie długość cyklu dla 22 wynosi 16. Dla danych liczb i oraz j zadanie polega na wyznaczeniu maksymalnej długości cyklu dla wszystkich liczb między i a j, włączając obie te liczby.

Wejście: W pierwszym wierszu mamy liczbę par liczb, do wczytania. W kolejnych, ciąg par liczb całkowitych i i j, po jednej parze w wierszu. Wszystkie liczby są mniejsze od 1 000 000 i większe od 0 oraz $i <= j$.

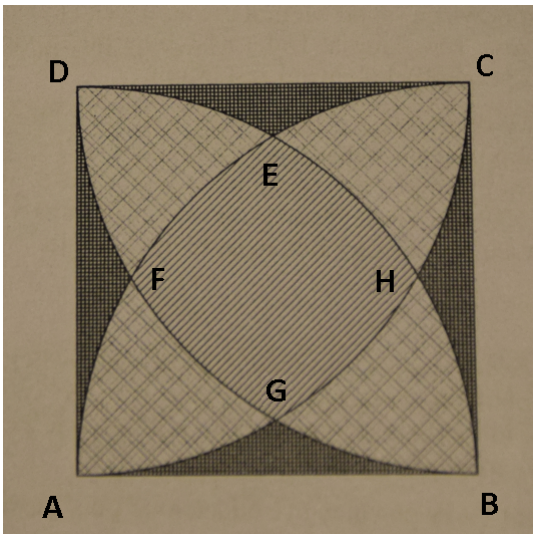
Wyjście: Dla każdej danej pary liczb i, j wynik ma zawierać liczby i, j w tej samej kolejności, w jakiej pojawiają się w danych wejściowych, a następnie maksymalną długość cyklu dla liczb od i do j włącznie. Te trzy liczby oddzielone spacjami (po jednej spacji między kolejnymi liczbami) powinny znajdować się w jednym wierszu, wierszy z kolei powinno być tyle, ile było wierszy w danych wejściowych, bez uwzględniania pierwszego wiersza.

Przykładowe dane	Wynik do przykładu
4 1 10 100 200 201 210 900 1000	1 10 20 100 200 125 201 210 89 900 1000 174

Zadanie4 - Czy to całkowanie?

Rysunek poniżej przedstawia kwadrat $ABCD$, w którym $AB = BC = CD = DA = a$. Narysowano cztery łuki o środkach w punktach A, B, C i D o promieniu a. Łuk będący fragmentem okręgu o środku A zaczyna się w sąsiednim wierzchołku B, a kończy się w drugim sąsiednim wierzchołku D. Inne łuki narysowane są w podobny sposób. W ten sposób tworzy się obszary o trzech różnych kształtach. Musisz określić pola obszarów: CDE, $EFGH$, CEH.

Wejście: Pierwszy wiersz zawiera liczbę wartości testowych. Każdy kolejny wiersz pliku z danymi zawiera liczbę zmiennopozycyjną a oznaczającą długość kwadratu, przy czym $0 \leqslant a \leqslant 10000$. Dane kończą się znacznikiem końca pliku.



Wyjście: Dla każdego zestawu testowego wypisz jeden wiersz zawierający wartości pól dla różnych typów obszarów. Każda liczba zmiennopozycyjna powinna być wypisana z dokładnością do trzech cyfr po przecinku. Pierwsza liczba w każdym zestawie powinna oznaczać pole obszaru $EFGH$, druga pole obszaru CEH, natomiast trzecia pole obszaru CDE.

Przykładowe dane	Wynik do przykładu
3 0.1 0.2 0.3	0.003 0.001 0 0.013 0.005 0.002 0.028 0.012 0.004

Zadanie5 - Gdzie jest Piotrek?

Znajdź położenie p słów w danej macierzy liter o wymiarach m na n. Dopasowanie może być wykonane w poziomie, w pionie oraz po przekątnych, w sumie osiem kierunków. Słowo znajduje się w macierzy, gdy kolejne jego litery należą do prostej nieprzerwanej linii pól macierzy. Przy dopasowaniu, wielkość liter nie ma znaczenia.

Wejście: Dane wejściowe rozpoczynają się od wiersza zawierającego jedną liczbę całkowitą dodatnią, oznaczającą liczbę zestawów danych. Każdy zestaw rozpoczyna się parą liczb całkowitych m i n w wierszu, przy czym $1 \leqslant m, n \leqslant 50$. Następnych m wierszy po n liter w każdym reprezentuje macierz, w której należy znaleźć słowa. W macierzy mogą znajdować się małe i wielkie litery. Po opisie macierzy w pliku następuje wiersz z jedną liczbą całkowitą k ($1 \leqslant k \leqslant 20$). Następnych k wierszy danych wejściowych zawiera listę słów do wyszukania, po jednym słowie w wierszu. Słowa te mogą zawierać jedynie małe i wielkie litery, nie dozwolone są spacje, myślniki i inne znaki spoza alfabetu.

Wyjście: Dla każdego słowa w każdym z zestawów testowych wynikiem powinna być para liczb całkowitych, przedstawiająca jego położenie w odpowiedniej macierzy. Pomiędzy tymi liczbami powinna być jedna spacja. Pierwsza z liczb mówi o tym, w którym wierszu macierzy znajduje się pierwsza litera szukanego słowa (1 to najwyższy wiersz, m to wiersz u samego dołu). Druga z tych liczb oznacza kolumnę macierzy, w której znajduje się pierwsza litera szukanego słowa (1 oznacza kolumnę najbardziej na lewo, a n-kolumnę najbardziej na prawo). Jeżeli słowo znajduje się więcej niż raz w macierzy, wynikiem powinna być pozycja najwyższego wystąpienia tego słowa w macierzy. Jeżeli najwyższe są dwa lub więcej słów, wynikiem powinno być wystąpienie najbardziej na lewo. Każde słowo występuje w macierzy co najmniej raz.

Wyniki dla dwóch kolejnych zestawów powinny być oddzielone pustym wierszem

Przykładowe dane wejściowe	Wynik do przykładu
1 8 11 *a b c D E F G h i N g* *h P b k W a I D S y k* *F o i A w a I t O t m* *F l s O m r o g s z c* *b s o A t m e D e s v* *K k c b i R i k o l k* *s a r L B G e d h O b* *y U i q l x c K B j f* 4 *Piotrek* *Polska* *Stomil* *Olsztyn*	2 2 2 2 2 9 7 10

Zadanie6 - Problem wyboru przeciwnika

Ród Targaryenów postanowił rozszerzyć terytoria wokół swoich zamków. W okolicy jego twierdz znajdują się grody Starków i Lannisterów. Aby uniknąć waśni z dwoma rodami na danym terenie, Targaryanie wybierają dla każdego zamku przeciwnika, dla którego suma odległości do najbliższych dwóch zamków jest najmniejsza, z drugim rodem zawiązują na danym terenie sojusz. Zadaniem jest określenie z którym rodem na danym terenie Targaryanie wchodzą w konflikt. W przypadku gdy sumaryczne odległości dwóch grodów do danego zamku Targaryenów będą identyczne dla obu rodów, wybieramy do walki Lannisterów.

Wejście:
wiersz 1 - liczba zamków Targaryenów,
wiersz 2 - x y jako współrzędne pierwszego zamku Targaryenów,
wiersz 3 - liczba współrzędnych grodów Lannisterów
wiersz 4 - współrzędne grodów Lannisterów
wiersz 5 - liczba współrzędnych grodów Starków
wiersz 6 - współrzędne grodów Starków
wiersz 7 - x y jako współrzędne drugiego zamku Targaryenów,
itd. aż do wyczerpania danych dla wszystkich zamków Targaryenów,

Wyjście: W wyniku dla każdego zamku Targaryenów wypisujemy w kolejnych wierszach nazwy przeciwników. W przypadku Starków wypisujemy fightStark, w przypadku Lannisterów wypisujemy fightLannister.

Przykładowe dane wejściowe	Wynik do przykładu
2 5 6 4 2 3 1 4 2 3 3 4 3 1 2 3 4 6 7 5 7 4 2 3 1 4 2 3 3 4 3 1 2 3 4 6 7	$fightStark$ $fightStark$