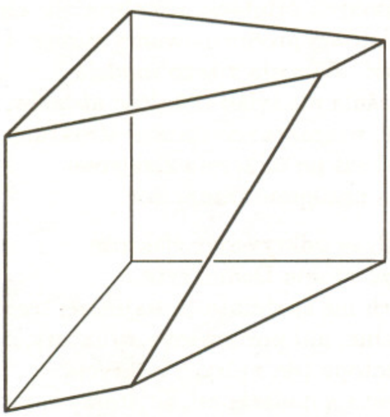
mała delta

Liczby i wielościany

Parę razy pisaliśmy w *Małej Delcie* o wielościanach, ale okazuje się, że prostych pytań na ich temat ciągle nie brakuje.

Pewna młoda dama dała mi do czytania tekst, w którym między innymi napisała: *Euler zauważył, że liczba ścian i wierzchołków nie wyznacza jednoznacznie wielościanu wypukłego. Postanowił więc dołączyć jeszcze trzecią liczbę – liczbę krawędzi.* Jest to, oczywiście, żart: liczba ścian S i liczba wierzchołków W wielościanu wypukłego wyznaczają jednoznacznie liczbę krawędzi K tego wielościanu; jest mianowicie $K = S + W - 2$.



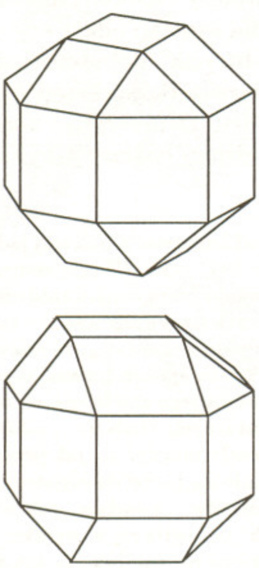
Rys. 1

Ale zatrzymajmy się na pierwszym zdaniu. Czy istotnie liczba ścian i liczba wierzchołków nie wyznaczają jednoznacznie wielościanu? Mamy tu dwa w jednym: pytanie *czy dla dowolnych* $S > 3$ *i* $W > 3$ *istnieje wielościan mający akurat taką liczbę ścian i taką liczbę wierzchołków* oraz pytanie *czy, jeśli* S *i* W *są, odpowiednio, liczbą ścian i wierzchołków jakiegoś wielościanu, to istnieje inny wielościan mający* S *ścian i* W *wierzchołków.*

Wypada doprecyzować słowo *inny*, bo, być może, od tego zależy odpowiedź. Będziemy więc tutaj uważać dwa wielościany za jednakowe, gdy różnić się będą jedynie długościami krawędzi i rozwartościami kątów. Tak więc bryła przedstawiona na rysunku 1 jest – w tym sensie – sześcianem.

Pierwsze z pytań pozostawimy tym razem do wyłącznej dyspozycji Czytelników (warto nadmienić, że niedawno w *Małej Delcie* znaleźć było można wskazanie wielościanu o danej z góry liczbie ścian, względnie wierzchołków). Tak więc będziemy tutaj rozważać jedynie pytanie, czy jeśli już wielościan istnieje, to nie ma innych o tej samej liczbie ścian i tej samej liczbie wierzchołków.

Zacznijmy od tego, że czasami rzeczywiście jest jedyność:
dla $S = W = 4$ jedynym wielościanem jest czworościan. A czy np. dla $S = 6$ i $W = 8$ istnieją jakieś wielościany oprócz tego z rysunku 1?



Rys. 2

Nie rozstrzygając tej kwestii zauważmy, że istnieją dwa różne wielościany o tych samych S i W, co więcej, oba archimedesowe (tzn. mające jednakowe naroża i ściany foremne). Są to wielościany, w których w każdym wierzchołku zbiegają się trzy czworokąty i jeden trójkąt (można sobie wyobrażać kwadraty i trójkąt równoboczny) – łącznie czworokątów jest 18, a trójkątów 8. Dla obu mamy $S = 26$, $W = 24$. To sławny przykład, bo podany dopiero w latach pięćdziesiątych naszego stulecia (wykrył go J. Załgaler) – do tego czasu jakoś nikomu nie wpadł w oko. Różnicę między tymi wielościanami zechce Czytelnik odczytać z rysunku 2.

MINISTERSTWO EDUKACJI NARODOWEJ
INSTYTUT INFORMATYKI UNIWERSYTETU WROCŁAWSKIEGO
KOMITET GŁÓWNY OLIMPIADY INFORMATYCZNEJ

V OLIMPIADA INFORMATYCZNA

Podstawowym aktem prawnym dotyczącym Olimpiady jest Regulamin Olimpiady Informatycznej, którego pełny tekst znajduje się w kuratoriach oświaty oraz jest zamieszczony w książce „IV Olimpiada Informatyczna 1996/97". Poniższe zasady są uzupełnieniem tego regulaminu, zawierającym szczegółowe postanowienia Komitetu Głównego Olimpiady Informatycznej o jej organizacji w roku szkolnym 1997/98.

Zasady organizacji zawodów w roku szkolnym 1997/98

§ 1. Wstęp

Olimpiada Informatyczna jest olimpiadą przedmiotową powołaną przez Instytut Informatyki Uniwersytetu Wrocławskiego, który jest organizatorem Olimpiady zgodnie z zarządzeniem nr 28 Ministra Edukacji Narodowej z dnia 14 września 1992 roku.

§ 2. Organizacja Olimpiady

1. Olimpiadę przeprowadza Komitet Główny Olimpiady Informatycznej.
2. Olimpiada Informatyczna jest trójstopniowa.
3. Olimpiada Informatyczna jest przeznaczona dla uczniów wszystkich typów szkół średnich dla młodzieży (z wyjątkiem szkół policealnych). W Olimpiadzie mogą również uczestniczyć – za zgodą Komitetu Głównego – uczniowie szkół podstawowych.
4. Integralną częścią rozwiązania każdego z zadań zawodów I, II i III stopnia jest program napisany na komputerze zgodnym ze standardem IBM PC, w jednym z następujących języków programowania: Pascal, C, lub C++.
5. Zawody I stopnia mają charakter otwarty i polegają na samodzielnym i indywidualnym rozwiązywaniu zadań i nadesłaniu rozwiązań w podanym terminie.
6. Zawody II i III stopnia polegają na indywidualnym rozwiązywaniu zadań w ciągu dwóch sesji przeprowadzanych w różnych dniach w warunkach kontrolowanej samodzielności.
7. Do zawodów II stopnia zostanie zakwalifikowanych 120 uczestników, których rozwiązania zadań I stopnia zostaną ocenione najwyżej; do zawodów III stopnia – 40 uczestników, których rozwiązania zadań II stopnia zostaną ocenione najwyżej. Komitet Główny może zmienić podane liczby zakwalifikowanych uczestników co najwyżej o 15%.
8. Podjęte przez Komitet Główny decyzje o zakwalifikowaniu uczestników do zawodów kolejnego stopnia, przyznanych miejscach i nagrodach oraz składzie polskiej reprezentacji na Międzynarodową Olimpiadę Informatyczną są ostateczne.
9. Terminarz zawodów:
   zawody I stopnia – 20 X 97r - 17 XI 97r.
   - ogłoszenie wyników 19 XII 97r.
   zawody II stopnia – 12-14 II 98r.
   - ogłoszenie wyników 2 III 98r.
   zawody III stopnia – 6-9 IV 98r.

§ 3. Wymagania dotyczące rozwiązań zadań zawodów I stopnia

1. Zawody I stopnia polegają na samodzielnym i indywidualnym rozwiązywaniu zadań eliminacyjnych (niekoniecznie wszystkich) i nadesłaniu rozwiązań pocztą, **przesyłką poleconą**, pod adresem:

Olimpiada Informatyczna
Ośrodek Edukacji Informatycznej i Zastosowań Komputerów
ul. Raszyńska 8/10
02-026 Warszawa
tel. (0-22) 822 40 19, 668 55 33

w nieprzekraczalnym terminie nadania do 17 listopada 1997 r. (decyduje data stempla pocztowego). Prosimy o zachowanie dowodu nadania przesyłki. Rozwiązania dostarczane w inny sposób nie będą przyjmowane.
2. Prace niesamodzielne lub zbiorowe nie będą brane pod uwagę.
3. Rozwiązanie każdego zadania składa się z:
   a) programu (tylko jednego) na dyskietce w postaci źródłowej i skompilowanej,
   b) opisu algorytmu rozwiązania zadania z uzasadnieniem jego poprawności.
4. Uczestnik przysyła jedną dyskietkę, oznaczoną jego imieniem i nazwiskiem, nadającą się do odczytania na komputerze IBM PC i zawierającą:
   – spis zawartości dyskietki w pliku nazwanym SPIS.TRC,
   – wszystkie programy w postaci źródłowej i skompilowanej.
   Imię i nazwisko uczestnika powinno być podane w komentarzu na początku każdego programu.
5. Wszystkie nadsyłane teksty powinny być drukowane lub czytelnie pisane na kartkach formatu A4. Każda kartka powinna mieć kolejny numer i być opatrzona pełnym imieniem i nazwiskiem autora. Na pierwszej stronie nadsyłanej pracy każdy uczestnik Olimpiady podaje następujące dane:
   – imię i nazwisko,
   – datę i miejsce urodzenia,
   – dokładny adres zamieszkania i ewentualnie numer telefonu,
   – nazwę, adres, województwo i numer telefonu szkoły oraz klasę, do której uczęszcza,
   – nazwę i numer wersji użytego języka programowania,
   – opis konfiguracji komputera, na którym rozwiązał zadania.
6. Nazwy plików z programami w postaci źródłowej powinny mieć jako rozszerzenie co najwyżej trzyliterowy skrót nazwy użytego języka programowania, to jest:

Pascal	PAS
C	C
C++	CPP

7. Opcje kompilatora powinny być częścią tekstu programu. Zaleca się stosowanie opcji standardowych.

§ 4. Uprawnienia i nagrody

1. Uczestnicy zawodów II stopnia, których wyniki zostały uznane przez Komitet Główny Olimpiady za wyróżniające, otrzymują najwyższą ocenę z informatyki na zakończenie nauki w klasie, do której uczęszczają.
2. Uczestnicy Olimpiady, którzy zostali zakwalifikowani do zawodów III stopnia, są zwolnieni z egzaminu dojrzałości (zgodnie z zarządzeniem nr 29 Ministra Edukacji Narodowej z dnia 30 listopada 1991 r.) lub z egzaminu z przygotowania zawodowego z przedmiotu informatyka. Zwolnienie jest równoznaczne z wystawieniem oceny najwyższej.
3. Laureaci i finaliści Olimpiady są zwolnieni w części lub w całości z egzaminów wstępnych do szkół wyższych na mocy uchwał senatów poszczególnych uczelni, podjętych zgodnie z przepisami ustawy z dnia 12 września 1990 roku o szkolnictwie wyższym (Dz.U. nr 65, poz. 385), o ile te uchwały nie stanowią inaczej.
4. Zaświadczenia o uzyskanych uprawnieniach wydaje uczestnikom Komitet Główny.
5. Komitet Główny ustala skład reprezentacji Polski na X Międzynarodową Olimpiadę Informatyczną w 1998 roku na podstawie wyników zawodów III stopnia i regulaminu tej Olimpiady. Szczegółowe zasady zostaną podane po otrzymaniu formalnego zaproszenia na X Międzynarodową Olimpiadę Informatyczną.
6. Nauczyciel (opiekun naukowy), który przygotował laureata Olimpiady Informatycznej, otrzymuje nagrodę przyznawaną przez Komitet Główny Olimpiady.
7. Uczestnicy zawodów II i III stopnia otrzymują nagrody rzeczowe.

MINISTERSTWO EDUKACJI NARODOWEJ
INSTYTUT INFORMATYKI UNIWERSYTETU WROCŁAWSKIEGO
KOMITET GŁÓWNY OLIMPIADY INFORMATYCZNEJ

V OLIMPIADA INFORMATYCZNA

§ 5. Przepisy końcowe

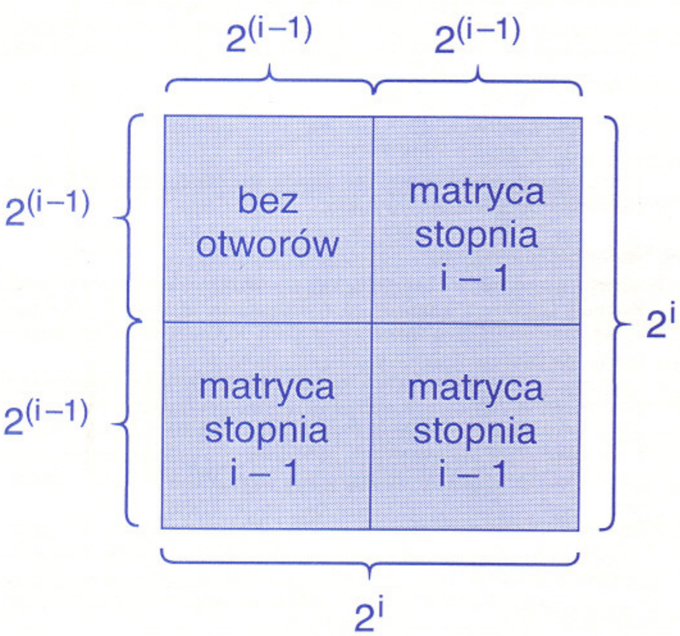
1. Koordynatorzy edukacji informatycznej i dyrektorzy szkół mają obowiązek dopilnowania, aby wszystkie informacje dotyczące Olimpiady zostały podane do wiadomości uczniów.
2. Komitet Główny Olimpiady Informatycznej zawiadamia wszystkich uczestników zawodów I i II stopnia o ich wynikach. Każdy uczestnik, który przeszedł do zawodów wyższego stopnia oraz dyrektor szkoły otrzymują informację o miejscu i terminie następnych zawodów.
3. Uczniowie zakwalifikowani do udziału w zawodach II i III stopnia są zwolnieni z zajęć szkolnych na czas niezbędny do udziału w zawodach, a także otrzymują bezpłatne zakwaterowanie i wyżywienie oraz zwrot kosztów przejazdu.

ZADANIA

PRACOWNIA MALARSKA

Pracownia malarska przygotowuje seryjną produkcję obrazów. Obrazy będą wykonywane za pomocą kwadratowych matryc o różnych stopniach. Matryca stopnia i składa się z 2^i wierszy i 2^i kolumn. Na przecięciu pewnych wierszy i kolumn znajdują się otwory. Matryca stopnia 0 ma jeden otwór.

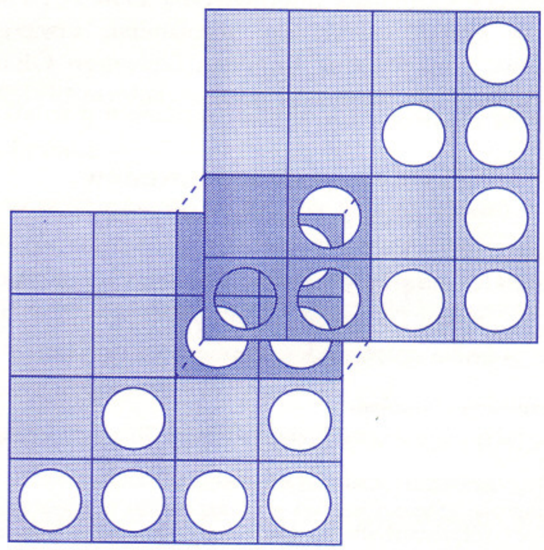
Dla $i > 0$, matryca stopnia i składa się z czterech kwadratów o rozmiarach $2^{(i-1)} \times 2^{(i-1)}$.



Oba prawe kwadraty oraz dolny lewy kwadrat są matrycami stopnia $i-1$. W górnym lewym kwadracie nie ma żadnych otworów. Obraz otrzymuje się w następujący sposób. Najpierw ustala się trzy nieujemne liczby całkowite n, x, y. Następnie umieszcza się dwie matryce stopnia n jedna na drugiej i górną matrycę przesuwa się o x kolumn w prawo i o y wierszy w górę. Tak otrzymany wzorzec zostaje umieszczony na białym płótnie i na wspólną część obu matryc nanosi się żółtą farbę. W efekcie na płótnie pojawią się żółte plamy tylko w tych miejscach, w których otwory w obu matrycach pokrywają się.

Przykład

Przyjrzyj się dwóm matrycom stopnia 2 przedstawionym na rysunku.



Górna matryca została przesunięta o 2 kolumny w prawo i o 2 wiersze w górę. W trzech miejscach otwory z obu matryc pokrywają się.

Zadanie

Napisz program, który:

- wczytuje z pliku tekstowego MAL.IN stopień obu matryc oraz współrzędne przesunięcia górnej matrycy;
- oblicza liczbę żółtych plam na płótnie;
- zapisuje wynik w pliku tekstowym MAL.OUT.

Wejście

Pierwszy wiersz pliku tekstowego MAL.IN zawiera liczbę całkowitą n, $0 \leq n \leq 100$. Jest to stopień matryc używanych w produkcji obrazów.

W drugim wierszu zapisana jest liczba całkowita x, zaś w trzecim wierszu liczba całkowita y, $0 \leq x, y < 2^n$. Liczba x jest liczbą kolumn, a y jest liczbą wierszy, o które należy przesunąć górną matrycę.

Wyjście

W pierwszym wierszu pliku wyjściowego MAL.OUT należy zapisać liczbę plam na płótnie.

Przykład

Dla pliku wejściowego MAL.IN:

```
2
2
2
```

poprawnym rozwiązaniem jest plik wyjściowy MAL.OUT:

```
3
```

Twój program powinien szukać pliku MAL.IN w katalogu bieżącym i tworzyć plik MAL.OUT również w bieżącym katalogu. Plik zawierający napisany przez Ciebie program w postaci źródłowej powinien mieć nazwę MAL.???, gdzie zamiast ??? należy wpisać co najwyżej trzyliterowy skrót nazwy użytego języka programowania. Ten sam program w postaci wykonalnej powinien być zapisany w pliku MAL.EXE.

MINISTERSTWO EDUKACJI NARODOWEJ
INSTYTUT INFORMATYKI UNIWERSYTETU WROCŁAWSKIEGO
KOMITET GŁÓWNY OLIMPIADY INFORMATYCZNEJ

V OLIMPIADA INFORMATYCZNA

WIELOKĄT

Powiemy, że dwa trójkąty **przecinają się**, jeśli ich wnętrza mają co najmniej jeden punkt wspólny. Wielokąt jest **wypukły**, jeśli każdy odcinek łączący dowolne dwa punkty tego wielokąta jest w nim zawarty.

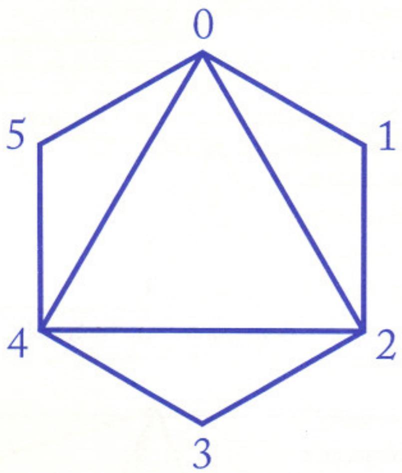
Trójkątem elementarnym w wielokącie wypukłym nazywamy każdy trójkąt, którego wierzchołkami są wierzchołki tego wielokąta.

Triangulacją wielokąta wypukłego nazywamy każdy zbiór elementarnych trójkątów w tym wielokącie, w którym żadne dwa trójkąty nie przecinają się, a wszystkie razem pokrywają cały wielokąt.

Dane są wielokąt wypukły i jego triangulacja. Jaka jest największa liczba trójkątów w tej triangulacji, które może przeciąć jeden elementarny trójkąt w tym wielokącie?

Przykład

Rozważmy następującą triangulację:



Trójkąt elementarny (1,3,5) przecina wszystkie trójkąty w tej triangulacji.

Zadanie

Napisz program, który:
- wczytuje z pliku tekstowego WIE.IN liczbę wierzchołków wielokąta i jego triangulację;
- oblicza największą liczbę trójkątów tej triangulacji, które przecina pojedynczy, elementarny trójkąt w danym wielokącie;
- zapisuje wynik w pliku tekstowym WIE.OUT .

Wejście

Pierwszy wiersz pliku WIE.IN zawiera liczbę n, $3 \leq n \leq 1000$. Jest to liczba wierzchołków wielokąta.

Wierzchołki wielokąta są ponumerowane kolejno 0, 1, ..., n-1, zgodnie z ruchem wskazówek zegara.

Kolejne n-2 wiersze zawierają opisy trójkątów w triangulacji. W wierszu i+1, $1 \leq i \leq n$-2, zapisane są trzy liczby całkowite a, b, c oddzielone pojedynczymi odstępami. Są to numery wierzchołków i-tego trójkąta w triangulacji.

Wyjście

W pierwszym i jedynym wierszu pliku WIE.OUT należy zapisać jedną liczbę całkowitą – największą liczbę trójkątów w triangulacji, które przecina jeden elementarny trójkąt w wielokącie wejściowym.

Przykład

Dla pliku wejściowego WIE.IN:

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

poprawnym rozwiązaniem jest plik tekstowy WIE.OUT:

```
4
```

Twój program powinien szukać pliku WIE.IN w katalogu bieżącym i tworzyć plik WIE.OUT również w bieżącym katalogu. Plik zawierający napisany przez Ciebie program w postaci źródłowej powinien mieć nazwę WIE.???, gdzie zamiast ??? należy wpisać co najwyżej trzyliterowy skrót nazwy użytego języka programowania. Ten sam program w postaci wykonalnej powinien być zapisany w pliku WIE.EXE.

SUMA CIĄGU JEDYNKOWEGO

Ciąg liczbowy o wartościach będących liczbami całkowitymi nazywamy **jedynkowym**, jeżeli dowolne jego sąsiednie wyrazy różnią się od siebie dokładnie o jeden oraz jego pierwszy wyraz jest równy 0. Bardziej precyzyjnie: niech $[a_1, a_2,...,a_n]$ będzie ciągiem o wartościach całkowitych; powiemy, że ten ciąg jest **jedynkowy**, jeżeli
- dla dowolnej liczby całkowitej k spełniającej nierówność $1 \leq k < n$ zachodzi warunek $|a_k - a_{k+1}| = 1$ oraz
- a_1=0.

Zadanie

Napisz program, który:
- wczyta z pliku tekstowego SUM.IN dwie liczby całkowite: długość ciągu i sumę elementów ciągu;
- wyznaczy ciąg jedynkowy o zadanej długości i sumie elementów lub stwierdzi, że taki ciąg nie istnieje;
- zapisze rezultat w pliku tekstowym SUM.OUT.

Wejście

W pliku tekstowym SUM.IN są zapisane:
- w pierwszym wierszu – liczba n elementów ciągu, spełniająca nierówność $1 \leq n \leq 10\,000$;
- w drugim wierszu – liczba S będąca żądaną sumą elementów ciągu, spełniająca nierówność $|S| \leq 50\,000\,000$.

Wyjście

W pierwszych n wierszach pliku tekstowego SUM.OUT należy zapisać n liczb całkowitych (po jednej w wierszu) będących kolejnymi wyrazami ciągu jedynkowego (k-ty wyraz w k-tym wierszu) o zadanej sumie S lub słowo NIE, jeżeli taki ciąg nie istnieje.

Przykład

Dla pliku wejściowego SUM.IN o zawartości:

```
8
4
```

poprawnym rozwiązaniem jest plik wyjściowy SUM.OUT o zawartości:

```
0
1
2
1
0
-1
0
1
```

Twój program powinien szukać pliku SUM.IN w katalogu bieżącym i tworzyć plik SUM.OUT również w bieżącym katalogu. Plik zawierający napisany przez Ciebie program w postaci źródłowej powinien mieć nazwę SUM.???, gdzie zamiast ??? należy wpisać co najwyżej trzyliterowy skrót nazwy użytego języka programowania. Ten sam program w postaci wykonalnej powinien być zapisany w pliku SUM.EXE.

MINISTERSTWO EDUKACJI NARODOWEJ
INSTYTUT INFORMATYKI UNIWERSYTETU WROCŁAWSKIEGO
KOMITET GŁÓWNY OLIMPIADY INFORMATYCZNEJ

V OLIMPIADA INFORMATYCZNA

AB-SŁOWA

Każdy niepusty ciąg, którego elementami są małe litery a i b, a także ciąg pusty nazywamy **ab-słowem**. Jeżeli $X = [x_1, ..., x_n]$ jest ab-słowem, a i, j takimi dowolnymi liczbami całkowitymi, że $1 \le i \le j \le n$, to przez $X[i..j]$ będziemy oznaczali podsłowo X składające się z kolejnych liter $x_i,...,x_j$. Powiemy, że ab-słowo $X = [x_1, ..., x_n]$ jest **ładnie zbudowane**, jeżeli zawiera tyle samo liter a co b i dla każdego i=1,...n podsłowo $X[1..i]$ zawiera co najmniej tyle samo liter a co liter b.

Podamy teraz indukcyjną definicję **podobieństwa** ładnie zbudowanych ab-słów:

- każde dwa puste ab-słowa (tzn. nie zawierające żadnych liter) są podobne,
- dwa niepuste, ładnie zbudowane ab-słowa $X = [x_1, ..., x_n]$ i $Y = [y_1, ..., y_m]$ są podobne, jeżeli są tej samej długości (n=m) i jest spełniony jeden z następujących warunków:
  1. x_1=y_1, x_n=y_n oraz $X[2..n-1]$ i $Y[2..n-1]$ są ładnie zbudowanymi, podobnymi ab-słowami,
  2. istnieje i, $1 \le i < n$, takie, że słowa $X[1..i]$, $X[i+1..n]$ są ładnie zbudowanymi ab-słowami i
     a) $Y[1..i]$, $Y[i+1..n]$ są ładnie zbudowanymi ab-słowami i $X[1..i]$ jest podobne do $Y[1..i]$ oraz $X[i+1..n]$ jest podobne do $Y[i+1..n]$, lub
     b) $Y[1..n-i]$, $Y[n-i+1..n]$ są ładnie zbudowanymi ab-słowami i $X[1..i]$ jest podobne do $Y[n-i+1..n]$ oraz $X[i+1..n]$ jest podobne do $Y[1..n-i]$.

Stopniem zróżnicowania niepustego zbioru S ładnie zbudowanych ab-słów nazywamy największą liczbę ab-słów, które można wybrać z S tak, żeby żadne dwa wybrane słowa nie były do siebie podobne.

Zadanie

Napisz program który:
- wczyta z pliku tekstowego ABS.IN elementy zbioru S;
- policzy stopień zróżnicowania S;
- zapisze wynik w pliku tekstowym ABS.OUT.

Wejście

W pliku tekstowym ABS.IN są zapisane :
- w pierwszym wierszu liczba n elementów zbioru S, $1 \le n \le 1000$;
- w kolejnych n wierszach elementy zbioru S tj. ładnie zbudowane ab-słowa, po jednym w każdym wierszu; pierwsza litera każdego ab-słowa jest pierwszym symbolem w wierszu i między kolejnymi literami w słowie nie ma żadnych innych znaków; każde ab-słowo ma długość co najmniej 1, a co najwyżej 200.

Wyjście

W pierwszym i jedynym wierszu pliku tekstowego ABS.OUT należy zapisać jedną liczbę całkowitą – stopień zróżnicowania S.

Przykład

Dla pliku wejściowego ABS.IN o zawartości:
```
3
aabaabbbab
abababaabb
abaaabbabb
```
poprawnym rozwiązaniem jest plik ABS.OUT o zawartości:
```
2
```

Twój program powinien szukać pliku ABS.IN w katalogu bieżącym i tworzyć plik ABS.OUT również w bieżącym katalogu. Plik zawierający napisany przez Ciebie program w postaci źródłowej powinien mieć nazwę ABS.???, gdzie zamiast ??? należy wpisać co najwyżej trzyliterowy skrót nazwy użytego języka programowania. Ten sam program w postaci wykonalnej powinien być zapisany w pliku ABS.EXE.

Wskazówki dla uczestników:
1. Przeczytaj uważnie nie tylko tekst zadań, ale i treść „Zasad organizacji zawodów".
2. Przestrzegaj dokładnie warunków określonych w tekście zadania, w szczególności wszystkich reguł dotyczących nazw plików.
3. Twój program powinien czytać dane z pliku i zapisywać wyniki do pliku. Nazwy tych plików powinny być takie jak podano w treści zadania.
4. Dane testowe są zawsze zapisywane bezbłędnie, zgodnie z warunkami zadania i podaną specyfikacją wejścia. Twój program nie musi tego sprawdzać. Nie przyjmuj żadnych założeń, które nie wynikają z treści zadania.
5. Staraj się dobrać taką metodę rozwiązania zadania, która jest nie tylko poprawna, ale daje wyniki w jak najkrótszym czasie.
6. Ocena za rozwiązanie zadania jest określona na podstawie wyników testowania programu i uwzględnia poprawność oraz efektywność metody rozwiązania użytej w programie.

Przewodniczący Komitetu Głównego
Olimpiady Informatycznej

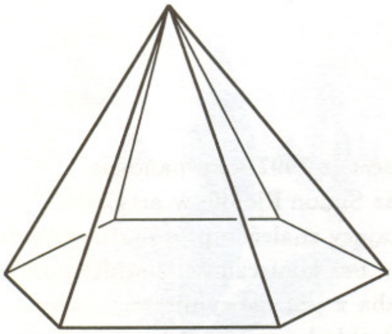
prof. dr hab. inż. Stanisław Waligórski

Olimpiada Informatyczna mogła dotąd działać dzięki wsparciu wielu osób, instytucji i firm. W przeprowadzeniu i uświetnieniu IV Olimpiady szczególnie pomogły:

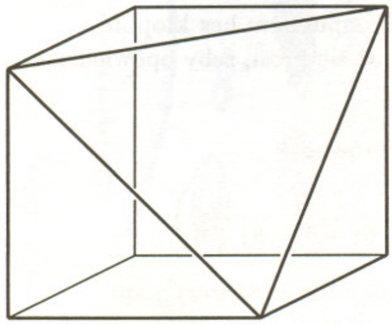
Polsko-Japońska Wyższa Szkoła Technik Komputerowych w Warszawie	udostępnienie budynku i świetnie wyposażonych pracowni komputerowych na zawody II i III stopnia oraz stworzenie bardzo dobrych warunków pracy w czasie zawodów
Ogólnopolska Fundacja Edukacji Komputerowej	ufundowanie cennych nagród dla laureatów i stałe wspieranie finansowe Olimpiady
Uniwersytet im. Mikołaja Kopernika w Toruniu	udostępnienie budynku i bardzo dobrze wyposażonej pracowni komputerowej na zawody II stopnia
Optimus SA	ufundowanie nagród dla najlepszych zawodników
Wydawnictwo Naukowe PWN	stałe obdarowywanie zawodników cennymi, wspierającymi ich rozwój książkami
Ośrodek Edukacji Informatycznej i Zastosowań Komputerów w Warszawie	wspieranie organizacyjne Olimpiady

i wielu pracowników i studentów Instytutów Informatyki Uniwersytetu Warszawskiego i Uniwersytetu Wrocławskiego

Polega ona na tym, że fragment tego wielościanu został obrócony względem pozostałej części o kąt 45°. Ale tego przykładu akurat Euler raczej nie znał. Formalnie więc odpowiedź jest **tak**, ale jej styl może się nie podobać.



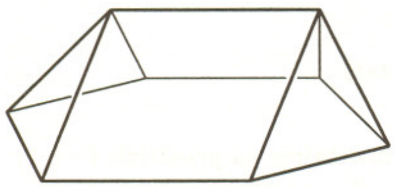
Rys. 3. $(W, K, S) = (7, 12, 7)$, w układzie $((1\times 6,\ 6\times 3),\ 12,\ (1\times 6,\ 6\times 3))$.



Rys. 4. $(W, K, S) = (7, 12, 7)$, w układzie $((3\times 4,\ 4\times 3),\ 12,\ (3\times 4,\ 4\times 3))$.

A czy są bardziej różniące się wielościany o tych samych S i W? Jak ich szukać? Metoda, którą proponuję, jest następująca. Podzielmy ściany według liczby boków. Pomnóżmy liczbę ścian danego rodzaju przez liczbę boków tej ściany i dodajmy wszystkie w ten sposób otrzymane iloczyny dla danego wielościanu. Otrzymana liczba to $2K$ i nietrudno się domyślić dlaczego. Można więc tę liczbę $2K$ rozkładać na takie iloczyny. Dla tych, którzy bardziej lubią wierzchołki od ścian, jest analogiczna propozycja – tym razem wierzchołki dzielimy według liczby wychodzących z nich krawędzi. Na rysunku 3 przedstawiony jest ostrosłup sześciokątny, odpowiadająca mu trójka $(W,\ K,\ S)$ i rozkład $2K$ na sumę iloczynów: jest jeden sześciokąt i 6 trójkątów $(1\times 6 + 6\times 3 = 24)$ – ciekawe, że jest to również rozkład na sumę iloczynów według wierzchołków (jest jeden sześcioramienny i 6 trójramiennych). Ale można rachować i tak: $24 = 3\times 4 + 4\times 3$ (ścian jest, tak jak poprzednio, 7) – temu rozkładowi odpowiada też wielościan o tych samych, co ostrosłup, wartościach S i W (rys. 4). A więc rzeczywiście odpowiedź na rozważane pytanie jest pozytywna: *istnieją zupełnie różne wielościany mające tę samą liczbę ścian, tę samą liczbę wierzchołków i tę samą liczbę krawędzi* (niematematyczne słowo *zupełnie* napisałem ze względu na przykład Załgalera).

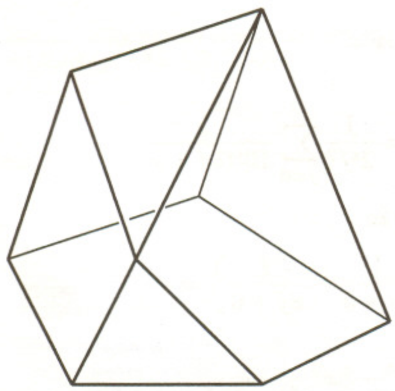
Można by uważać, że to jest koniec problemu. Ale dla usposobienia matematycznego będzie to właśnie początek. Rozważmy pytanie, czy każdemu rozkładowi odpowiada jakiś wielościan. Dla samodzielnej pracy Czytelnika proponuję rozważenie bardzo podobnego przykładu – ostrosłupa ośmiokątnego. Dla niego mamy
$1\times 8 + 8\times 3 = 32 = 5\times 4 + 4\times 3;$
czy istnieje drugi z tych wielościanów?



Rys. 5. $(W, K, S) = (8, 13, 7)$, w układzie $((2\times 4,\ 6\times 3),\ 13,\ (1\times 6,\ 2\times 4,\ 4\times 3))$.

Polecam też rozwiązany przeze mnie przykład namiotu o siedmiu ścianach z rysunku 5. Mamy:

$$26 = 1\times 6 + 2\times 4 + 4\times 3 = 1\times 5 + 3\times 4 + 3\times 3 = 2\times 5 + 1\times 4 + 4\times 3 = \\ = 1\times 7 + 1\times 4 + 5\times 3 = 1\times 8 + 6\times 3.$$

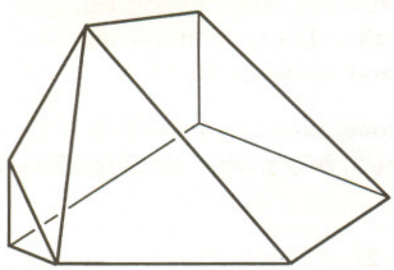


Rys. 6. $(W, K, S) = (8, 13, 7)$, w układzie $((2\times 4,\ 6\times 3),\ 13,\ (1\times 5,\ 3\times 4,\ 3\times 3))$.

Tutaj wielościany odpowiadające rozkładom podanym w pierwszym wierszu istnieją (rys. 5–7), a odpowiadające rozkładom w drugim wierszu – nie. Nie jest tym razem trudno uzasadnić, dlaczego ostatnie dwa rozkłady są złe (np. w ostatnim do ośmiokąta możemy przyczepić jedynie sześć innych ścian, co wielościanu nie utworzy).

I znów powstają pytania:
- jak poznać, którym rozkładom odpowiadają wielościany, a którym nie?
- ilu maksymalnie rozkładom danego S odpowiadają wielościany (widać, że mogą być trzy, ale może ta liczba rośnie wraz z S) ?
- które z wielościanów mają swoich załgalerowskich krewnych, czyli wielościany inaczej złożone z tych samych ścian?

No i, oczywiście, wiele innych pytań.



Rys. 7. $(W, K, S) = (8, 13, 7)$, w układzie $((2\times 4,\ 6\times 3),\ 13,\ (2\times 5,\ 1\times 4,\ 4\times 3))$.

Małą Deltę przygotował Marek KORDOS