POMORSKA LIGA ZADANIOWA *ZDOLNI Z POMORZA*

Konkurs dla uczniów szkół ponadpodstawowych i ponadgimnazjalnych województwa pomorskiego w roku szkolnym 2020/2021

Etap III-wojewódzki

Przedmiot: Informatyka

Czas przewidziany na rozwiązanie zadań konkursowych: 12:00 – 13:30 (90 minut)
Dodatkowy czas przewidziany na czynności organizacyjne: 13:30 – 13:45 (15 minut)

Przed przystąpieniem do rozwiązywania zadań zapoznaj się z instrukcją.

Instrukcja dla rozwiązującego

1. Arkusz składa się z 15 stron i zawiera 5 zadań (numerowane od 1 do 5).
2. **Za rozwiązanie wszystkich zadań możesz uzyskać maksymalnie 50 punktów.**
3. Do zadania 1 otrzymujesz dodatkowo 2 załączniki ***Zadanie1-Zalacznik1-projekty.txt*** oraz ***Zadanie1-Zalacznik1-wspolczynniki.txt.*** Przed przystąpieniem do rozwiązywania sprawdź, czy na pewno pobrała(e)ś te pliki. Nie wolno używać własnych plików zamiast tych załączników.
4. W zadaniach 4 oraz 5 pliki o charakterze testowym możesz tworzyć samodzielnie. Nie przekazuj ich jednak do oceny.
5. Rozwiązania zadań 1-5 umieszczaj w plikach o dokładnie takich nazwach, jak określono w treści zadań. Nie przekazuj do oceny innych plików niż te określone w treści zadań.
6. **Po zakończeniu pracy wszystkie pliki zawierające rozwiązania zadań umieść w folderze nazwanym swoim nazwiskiem, spakuj ten folder (np. formaty plików zip, rar lub z) i prześlij ten plik-archiwum na adres wskazany przez organizatorów. Jeśli Twój program pocztowy nie pozwoli Ci wysłać pliku**

spakowanego to udostępnij go za pośrednictwem dostępnych serwisów, w taki sposób, aby organizatorzy mogli go bez problemu pobrać, a potem prześlij organizatorom na wskazany adres dokładną informację na ten temat.

7. **W treści wysyłanego listu elektronicznego podaj swoje dane identyfikacyjne tzn. imię, nazwisko, nazwę szkoły i adres szkoły. Uczestnik zobowiązany jest podać tytuł wiadomości zgodny z wzorem: np. Kowalski Jan – informatyka – szkoła – ponadpodstawowa.**
8. **Czas przeznaczony na odbiór treści zadań i załączników, lekturę instrukcji oraz przygotowanie i wysłanie rozwiązań nie jest zaliczany do czasu przeznaczonego na rozwiązanie zadań. Czas przeznaczony na te czynności wynosi: 15 minut.**
9. ROZWIĄZANIA ZADAŃ NALEŻY PRZESŁAĆ NA ADRES: **informatyka_plz_Pp@odn.slupsk.pl**
10. Przy rozwiązywaniu zadań wykorzystujesz dostępne narzędzia i środowiska. Powinny one odpowiadać środowiskom i narzędziom, które były dopuszczone do użytku podczas etapu powiatowego.
11. Dla rozwiązań (zwłaszcza w językach programowania) obowiązują te same zasady szczegółowe, jak w trakcie etapu powiatowego:

a) dopuszcza się używanie wyłącznie standardowej biblioteki (bibliotek) języka, nie jest dozwolone dołączanie zewnętrznych bibliotek np. crt, graph (poza sytuacjami wynikłymi z treści zadania np. próba realizacji rysunku wynikającego z treści zadania);

b) nie jest dopuszczalne otwieranie przez program innych programów, plików (poza tymi z danymi wejściowymi i wyjściowymi), ani tworzenie nowych plików (np. tymczasowych) oraz tworzenie innych procesów lub wątków;

c) błąd kompilacji przy sprawdzaniu jest traktowany jako błąd składni i sprawdzający nie ma obowiązku dalszej analizy takiego rozwiązania choć może uwzględnić poprawny zarys samego algorytmu przyznając znacząco mniejszą liczbę punktów. Podobna uwaga dotyczy pojawienia się nietypowych błędów wykonania w trakcie uruchamiania programów (np. naruszenie zasad ochrony pamięci);

d) rozwiązania nie powinny wykorzystywać plików nagłówkowych typowych dla środowisk DOS/Windows np. conio.h lub windows.h (dotyczy języka C++);

e) rozwiązania nie powinny naruszać bezpieczeństwa systemowego w środowisku, w którym są sprawdzane.

12. Programy nie powinny zajmować się testowaniem poprawności danych. Zakłada się, że ma ona miejsce.
13. Proszę zwrócić uwagę na samodzielność rozwiązań. **Mimo, że pracujesz w warunkach nie w pełni kontrolowanej samodzielności pamiętaj, że w każdej rywalizacji ważne są zasady fair play.**

Powodzenia !

Zadanie 1

Inżynieria oprogramowania to dział informatyki zajmujący się najogólniej tematyką tworzenia oprogramowania. Jednym z problemów w tej dziedzinie jest kwestia wyceny kosztów realizacji projektów związanych ze stworzeniem określonych systemów informatycznych. Istnieje różne modele tej wyceny, a niektóre z nich podają nawet konkretne formuły obliczeniowe z nią związane.

Oto jeden z nich. Tzw. kosztochłonność (jej jednostką są tzw. osobomiesiące, ale tym się nie będziemy tu zajmować) projektu K jest obliczana ze wzoru:

$K=a\ (DK)^E$

Przy czym DK oznacza długość kodu oprogramowania mierzoną liczba napisanych przez programistów wierszy zaś *a* jest stałą, która przyjmuje się do obliczeń zależnie od typu projektu.

Z kolei:

$E=b+0.01*(W1+W2+W3+W4)$

gdzie *b* to kolejna przyjmowana zależnie od typu projektu stała, zaś 4 występujące w nawiasie współczynniki dotyczą pewnych charakterystyk projektu i są dobierane z tabel:

W1- wiąże się z tzw. typowością projektu (bardziej typowy, czy bardziej innowacyjny),

W2-wiąże się z tzw. elastycznością projektu (nie rozwijamy tematu),

W3- wiąże się z zarządzaniem ryzykiem przy realizacji projektu,

W4- wiąże się z tzw. spójnością zespołu realizującego projekt (dotyczy współpracy w zespole).

W zakresie każdej z tych charakterystyk projekt może należeć do jednej z kategorii numerowanych od 1 do 5 (kategoria może być dla każdej charakterystyki inna np. 1 dla typowości, ale 4 dla elastyczności itp.) , a w zależności od tego do której ze względu na daną charakterystykę należy wartości współczynników W1,W2,W3 oraz W4 dobiera się z poniższej tabeli:

Kategoria	Typowość(W1)	Elastyczność(W2)	Zarządzanie ryzykiem(W3)	Spójność zespołu(W4)
1	6,2	5,07	7,07	5,48
2	4,96	4,05	5,65	4,38
3	3,72	3,04	4,24	3,29
4	2,48	2,03	2,83	2,19
5	1,24	1,01	1,41	1,1

W zasadzie pozostaje tylko znając *DK* i przynależność projektu do poszczególnych kategorii w ramach kolejnych charakterystyk dobrać współczynniki *W1,..W4*, dobrać stałe *a* i *b* oraz podstawić wszystkie dane do wzoru i uzyskać wielkość K, ale wspomnijmy o pewnych trudnościach obliczeniowych:

a) długość kodu DK nie zawsze jest znana dokładnie w momencie tworzenia wyceny. Częściej znamy tzw. widełki czyli szacowaną najmniejszą długość kodu i szacowaną największą długość kodu. W takim przypadku wylicza się dwie wartości K dla tych dwóch długości kodu, a ostateczną wartością wyceny jest średnia z tych dwóch wartości K,

b) często trudno przypisać projekt do określonej kategorii, najtrudniej w zakresie elastyczności. Mówi się wówczas np. kategoria 2lub3 (2l3). Jaką wartość współczynnika wtedy wybrać ? Średnią ze współczynników dla kategorii 2 l 3. Podobnie dla innych współczynników podawanych w wersji „to lub to".

Dodajmy jeszcze ważne dla obliczeń informacje:

a) znając K można obliczyć tzw. minimalną liczbę osób niezbędną do realizacji projektu P ze wzoru: $P = K/(2.5*(K)^c)$

gdzie c to już ostatnia w tych obliczeniach stała przyjmowana zależnie od typu projektu.

b) pora też określić stałe, które wykorzystasz w tym zadaniu

a=2,94 b=0,91 c=0,71.

W załączonym pliku ***Zadanie1-Zalacznik1-projekty.txt*** znajduje się informacja o 10 projektach zaplanowanych do realizacji w najbliższym (pewnie kilkuletnim) okresie przez pewną firmę informatyczną. W kolejnych wierszach umieszczonych pod wierszem tytułowym znajdziesz oddzielone znakiem tabulacji następujące dane: *numer porządkowy projektu, oszacowaną minimalną długość kodu dla tego projektu, oszacowaną maksymalną długość kodu dla tego projektu, kategorię projektu w zakresie typowości, kategorię projektu w zakresie elastyczności, kategorię projektu w zakresie zarządzania ryzykiem, kategorię projektu w zakresie spójności zespołu.* Wszystkie kategorie są liczbami z zakresu 1-5, wyjątkiem jest kategoria elastyczność. W tek kategorii znajdziesz dla niektórych projektów zapisy typu 3l4 lub 4l5 (tylko takie, nie ma wariantów 1l2, ani 2l3). Intepretujemy je jako opisaną wyżej możliwość kategoria „taka lub taka" i postępujemy zgodnie z opisem w punkcie b mówiącym o dodatkowych trudnościach obliczeniowych.

Uwzględniając podane wyżej wzory, wartości stałych oraz inne informacje rozwiąż następujące problemy:

1. Dla każdego projektu oblicz jego wartość K (wycenę) oraz P (minimalną niezbędną liczbę pracowników do jego realizacji). Uwaga ! Wartości współczynników W1,W2,W3 oraz W4, które należy dobrać do obliczeń zależnie od kategorii przyznanej projektowi w zakresie danej charakterystyki znajdziesz w załączniku ***Zadanie1-Zalacznik1-wspolczynniki.txt.***
2. Biorąc pod uwagę, że rozważana firma zatrudnia na stałe 12 osób pokaż na wykresie ile osób brakuje do realizacji każdego projektu uwzględniając obliczone dla niego P.
3. Firma rozważa wariant przyspieszenia prac nad projektami poprzez połącznie ich w transze projektów, które będą realizowane równolegle(jednocześnie) z wykorzystaniem zatrudnionych na zlecenie dodatkowych podwykonawców. W pierwszej transzy będą realizowane równolegle pierwsze trzy projekty, po jej zakończeniu w drugiej transzy kolejne 3, a po jej zakończeniu w trzeciej transzy ostatnie 4 projekty. W każdej transzy przy realizowanych w niej projektach będą pracować jednocześnie nie tylko stali pracownicy firmy, ale i zatrudnieni podwykonawcy. Ci sami wykonawcy będą następnie przechodzić do pracy (podobnie jak wszyscy stali pracownicy) w kolejnej transzy. Oblicz z iloma maksymalnie podwykonawcami trzeba podpisać umowy na okres realizacji wszystkich transz (zakładamy, że umowa z

podwykonawcą jest podpisywana na okres od pierwszej do ostatniej transzy bez względu na to, czy wszyscy podwykonawcy w każdej transzy są potrzebni).

4. Wszystkie wyniki obliczeń istotnie zależą od trzech stałych *a, b, c* i zmieniają się przy zmianie ich wartości. Ponieważ stałe te zostały narzucone i zależnie od typu projektu można je dobierać inaczej. Ustal symulacyjnie dla jakiej mniejszej od obecnie ustalonej, ale możliwie największej wartości stałej *b* zachowując warunki realizacji projektu z punktu 3 nie będzie w ogóle potrzebne zatrudnianie podwykonawców. Wartości innych stałych i współczynników pozostają w tym problemie bez zmian. Wartość *b* należy podać z dokładnością do 2 miejsc po przecinku.

Do oceny oddajesz plik zawierający komputerową realizację obliczeń, na podstawie których uzyskasz rozwiązanie zadania. Nazwa tego pliku to *Zadanie1*. Jeśli takich plików jest więcej to kolejne mają nazwy *Zadanie1a, Zadanie 1b* itd. Dodatkowo wyniki liczbowe dotyczące problemów 1, 2, 4 oraz 5 umieść w pliku *Zadanie1-wyniki*.

Uwaga ! Odpowiedzi liczbowe umieszczone w pliku tekstowym *Zadanie1-wyniki* nie będą mogły być uznane nawet jeśli będą poprawne, o ile nie znajdą potwierdzenia i odzwierciedlenia w zawartości pliku z komputerową realizacją obliczeń.

11 punktów

Zadanie 2

Algorytm Huffmana jest wykorzystywany w kompresji danych. Jedną z jego cech charakterystycznych jest fakt, że kod proponowany dla poszczególnych znaków jest kodem prefiksowym tzn. kod żadnego znaku nie może być przedrostkiem (prefiksem) kodu innego znaku. Pozwala to stosunkowo łatwo odczytywać zakodowany ciąg znaków znając tylko ich kody Huffmana, bez obawy, że pomylimy pewien znak z innym znakiem. Przekonaj się o tym rozwiązując najpierw część a) tego zadania.

a) przypuśćmy, że przy pomocy algorytmu Huffmana uzyskano następujące kody dla poniższych 5 znaków (przypomnijmy, że kody Huffmana dla znaków nie są określone jednoznacznie – ma to miejsce tylko dla pełnego zestawu znaków ASCII, w przeciwnym razie kod zależy od tego jaki zbiór znaków kodujesz i jakie znaki się w nim znajdują):

s-00

e-01

a-11

t-100

r-101

Stosując te wartości kodów odczytaj jakie słowo zostało poniżej zakodowane:

001000110110011

b) wykorzystując swoje doświadczenia z odczytywaniem tekstu z punktu a) w dowolnej notacji (lista kroków, pseudokod, schemat blokowy) zapisz algorytm o następującej specyfikacji:

Dane: *n*-liczba naturalna większa od 1,

indeksowana struktura *x* zawierając *n* par (znak, kod Huffmana) – w każdej parze podano jeden ze znaków, który należy do tzw. drukowalnych znaków kodu ASCII oraz odpowiadający mu kod Huffmana uzyskany przy kodowaniu właśnie tych *n* znaków (znaki nie powtarzają się). Przyjmujemy, że najkrótsze kody znaków mają długość 2. Uwaga do poszczególnych elementów tej struktury można się w zapisie algorytmu odwoływać wg schematu *x[i].znak, x[i].kod* (i-numer pary od 1 do *n*),

t- dowolny tekst zakodowany przy pomocy kodu Huffmana utworzonego dla tej rodziny n znaków.

Wynik:

odkod- jawny tekst złożony z drukowalnych kodów ASCII odpowiadający zakodowanemu kodem Huffmana tekstowi *t* .

Dodatkowe uwagi:

a) w zapisie algorytmu można wykorzystać następujące dwie funkcje:

dł(t)- funkcja wyznaczająca długość tekstu *t*

sk(skad, od,ile) - funkcja zwracająca tekst o liczbie znaków *ile*, skopiowanych od miejsca *od* z tekstu o nazwie *skąd*. Jeżeli wartość *od* jest większa niż długość tekstu to funkcja zwraca tekst pusty. Jeżeli trzeciego parametru nie podano to funkcja zwraca wszystkie znaki tekstu *skąd* począwszy od znaku o numerze *od* do końca tekstu.

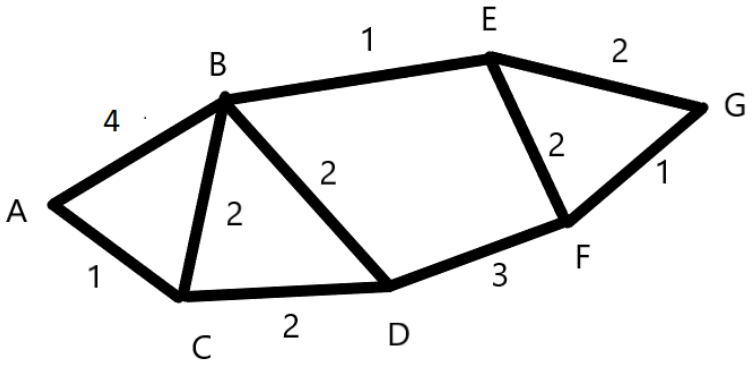
b) przyjmijmy, że znaki w tekstach są numerowane od 0.

Do oceny oddajesz plik tekstowy zawierający rozwiązania problemów przedstawionych w punkcie a oraz w punkcie b tego zadania. Nazwa tego pliku to *Zadanie2*

7 punktów

Zadanie 3

Poniżej przedstawiono schemat połączeń w pewnej sieci komputerowej.



Literami A, B oraz C zaznaczono źródła – tylko one mogą wysyłać pakiety z danymi, choć przez nie też może przechodzić ruch sieciowy. Z kolei G to komputer docelowy czyli przeznaczenie wszystkich wysyłanych pakietów, sam nie wysyła w rozważanym przykładzie żadnych pakietów danych . Pozostałe elementy oznaczone literami D, E oraz F to routery pośredniczące w przekazywaniu pakietów do kolejnych „odcinków" trasy, one same też nie wysyłają żadnych pakietów. Cyfry naniesione przy rożnych odcinkach to szacowany czas przejścia danego odcinka przez pakiet danych (jednostka czasu nie ma w tym przykładzie znaczenia).

W sieci obowiązują następujące zasady ruchu pakietów:

a) pakiety są wysyłane dokładnie co jedną jednostkę czasu, ale nie ma możliwości, aby w danej chwili czasu został wysłany więcej niż jeden pakiet- pakiet może „wyjść" oczywiście albo z A, albo z B, albo z C,

b) dla pakietu wychodzącego z danego źródła, który ma dotrzeć do G ustalana jest trasa o najkrótszym czasie trwania i pakiet stara się po niej poruszać (może wybrać dowolną z nich, gdy jest więcej tras o takim samym najkrótszym czasie „dotarcia" do G). Może ja zmienić tylko wtedy, gdy wystąpią przyczyny podane w punkcie c). Wtedy od miejsca, w którym musi zmienić trasę do G jest wyznaczana nowa trasa, najkrótsza (ze względu na czas trwania) z tych, które z danego punktu (zmiany trasy) są dla pakietu możliwe,

c) pakiet zmienia trasę tylko wtedy, gdy:

- znajdzie się w danym punkcie wraz z innymi pakietami, które też miałyby „wyruszyć" na ten sam odcinek, ale ma niższy priorytet niż inny pakiet lub pakiety. Priorytet jest wyznaczany na podstawie chwili uruchomienia wysyłki pakietu. Najwyższy priorytet ma zawsze ten pakiet , który został wysłany najwcześniej, a potem kolejno te, który były wysyłane w następnych jednostkach czasu. Z kilku pakietów „rywalizujących" o dany odcinek wyruszy zatem na niego ten o najwyższym priorytecie- inne muszą zmienić trasę.
- na odcinku, którym miałby za chwilę się poruszyć pozostaje jeszcze inny pakiet,

d) pakiet nie może pokonywać żadnego odcinka więcej niż jeden raz, ale może wielokrotnie przechodzić przez te same punkty,

e) jeśli pakiet znajdzie się podczas swojej trasy w punkcie, w którym w zgodzie z opisanymi zasadami nie będzie mógł wybrać żadnego odcinka to jest tracony.

Przypuśćmy, że mamy do czynienia z następującą sekwencją wysyłki pakietów

- pakiet nr 1 w chwili T=0 „wyrusza" z A
- pakiet nr 2 w chwili T=1 „wyrusza" z A
- pakiet nr 3 w chwili T=2 „wyrusza" z C
- pakiet nr 4 w chwili T=3 „wyrusza" z B

Opierając się na opisanych zasadach ruchu w prezentowanej sieci opisz drogę każdego z 4 pakietów do komputera docelowego G. Przez opis drogi każdego pakietu rozumiemy listę punktów przez, które wiodła trasa pakietu aż do osiągniecia punktu G, a przy każdym punkcie należy podać numer chwili (jednostki) czasu, w której się w tym punkcie znalazł- wszystkie czasy podajemy względem chwili T=0 tzn. jeśli pakiet wyruszył w chwili T=1, a po 2 jednostkach czasu znalazł się w określonym punkcie to jako czas podajemy 3, a nie 2,

Do oceny oddajesz plik tekstowy zawierający rozwiązania problemów przedstawionych w punkcie a oraz w punkcie b tego zadania. Nazwa tego pliku to *Zadanie3*.

9 punktów

Zadanie 4

Przyjmijmy kilka definicji dotyczących danych znakowych:

Parą prawidłową będziemy nazywać dwa znaki, z których pierwszy jest samogłoską (nie rozważamy w tym przykładzie polskich znaków więc chodzi o: A, a, E, e, I, i, O, o, U, u oraz Y i y), a drugi dowolnym znakiem zapisywanym inną niż samogłoska literą wielką lub mała. Przykłady par prawidłowych: yB, as, AT, Iv.

Odległością dwóch liter będziemy nazywać wartość bezwzględną z różnicy ich kodów ASCII, dla A i a jest to na przykład liczba |65-97| =32,

Tekst złożony wyłącznie z małych i dużych liter (nawet nie ma w nim spacji) będziemy nazywać **prawidłowym** jeżeli z uwzględnieniem określonych reguł (o jakie operacje chodzimy podajemy dalej) można na jego podstawie (czyli tylko ze znaków go tworzących) utworzyć nowy tekst, który będzie składał się z dwóch tylko części występujących koniecznie we wskazanej kolejności: pierwszą (koniecznie niepustą) będą tworzyły umieszczone jedna po drugiej pary prawidłowe, a drugą (ta może być pusta) litery nie tworzące par prawidłowych, które nie mogą być jednak samogłoskami.

Reguły, których należy przestrzegać, aby utworzyć z tekstu nową jego postać dowodzącą, że jest on tekstem prawidłowym są następujące:

a) nie wolno rozdzielać już istniejących w tekście par prawidłowych, można je natomiast w całości (czyli oba znaki pary) ustawić w dowolnym miejscu nowego tekstu,

b) dla innych samogłosek w tekście dobór znaków, które mogą z nimi utworzyć pary prawidłowe przebiega następująco: niewykorzystane (czyli nie pozostające w oryginalnym tekście w parach prawidłowych) samogłoski-kandydatki do kolejnych par prawidłowych rozważamy w kolejności ich występowania w tekście oryginalnym, a znak do pary prawidłowej dla każdej samogłoski dobieramy z innych pozostających do dyspozycji w tekście (i nie będących naturalnie samogłoskami) w taki sposób , aby odległość między samogłoską, a

dobieranym znakiem była najmniejsza z możliwych (gdy taka odległość jest równa dla kilku znaków-kandydatów do pary prawidłowej, to można wybrać do pary prawidłowej dowolny z nich). Tak utworzone pary prawidłowe możemy umieścić w dowolnym miejscu nowego tekstu.

Naturalnie jeśli w oryginalnym tekście pozostaną tylko samogłoski, z którymi nie da się już utworzyć par prawidłowych to ten tekst nie może być uznany za prawidłowy, a jeśli po utworzeniu możliwych par prawidłowych nie ma tam już znaków lub pozostały inne znaki niż samogłoski to mogą uzupełnić one nowy tekst dowodząc prawidłowości tekstu oryginalnego.

Przykład:
Tekst *albbcaedKEc* jest prawidłowy.
Budując nowy tekst możemy na jego początku ustawić istniejące już pary w tekście oryginalnym prawidłowe czyli *al,ed* i *Ec*. Teraz dla kolejnych samogłosek (ale jest już tylko jedna - a !) dobierzemy spośród pozostałych znaków te, które wg opisanych wyżej zasad utworzą z nimi pary prawidłowe. Dla *a* (bo od niego zgodnie z kolejnością występowania w tekście zaczynamy i tak jest zresztą jedyne) będzie to oczywiście *b* (jedna z liter b ściśle) . Te dwie nowe pary prawidłowe możemy dołączyć do już istniejących otrzymując pierwszą część poszukiwanej postaci tekstu:
aledEcab,

a potem dopiszemy do tego tekstu pozostałe niewykorzystane przy tworzeniu par prawidłowych znaki (czyli w kolejności *b, c, K* -choć kolejność tych liter jest akurat bez znaczenia) i ponieważ żądna z nich nie jest samogłoską więc pożądana postać tekstu może wyglądać np. w ten sposób (naturalnie kolejność par prawidłowych, jak i znaków w drugiej części otrzymanego tekstu może być dowolnie inna):
aledEcabbcK
czyli tekst pierwotny na podstawie, którego zbudowaliśmy ten nowy jest prawidłowy.
Twoje zadanie polegać będzie na tym, aby napisać program, który potrafi ocenić czy w ciągu na tekstów są teksty prawidłowe.

Dane wejściowe

Plik tekstowy ***teksty.txt*** zawierający w pierwszym wierszu liczbę naturalną **n** oznaczającą ilość tekstów w kolejnych liniach plików (n jest równe przynajmniej 1, ale nie jest większe niż 40). W każdym z tych kolejnych wierszy zapisany jest dokładnie jeden tekst o długości przynajmniej 2 i nieprzekraczającej 40 znaków złożony wyłącznie z małych i wielkich liter.

Dane wyjściowe

Plik tekstowy ***rezultaty.txt,*** składający się z *n* linii. W linii numer *i* tego pliku znajduje się albo jedna z możliwych postaci tekstu z wiersza numer *i+1* pliku ***teksty.txt,*** którą otrzymano z tekstu oryginalnego stosując reguły opisane w zadaniu (co dowodzi, że tekst z wiersza i+1 pliku ***teksty.txt*** jest tekstem prawidłowym) , albo liczba 0 jeżeli utworzenie postaci tekstu z linii i+1 pliku ***teksty.txt*** dowodzącej, że jest on tekstem prawidłowym nie jest możliwe.

Przykład

Jeżeli w pliku ***teksty.txt*** mamy następujące dane:

4
Alak
akeuyert
albbcaedKEc
abcdefiu

to w pliku ***rezultaty.txt*** powinniśmy dla przykładu otrzymać:

Alak
0
aledEcabbcK
abefiduc

Komentarz: w drugim tekście jest zbyt wiele samogłosek nie może być zatem na jego podstawie utworzona dowodząca jego prawidłowości postać, w pozostałych przypadkach

teksty są prawidłowe i otrzymano jedną z możliwych postaci o tym świadczących z uwzględnieniem opisanych w zadaniu reguł.

Do oceny oddajesz plik zawierający kod źródłowy napisanego przez Ciebie programu. Nazwa tego pliku to *Zadanie4.*

12 punktów

Zadanie 5

Niektóre liczby naturalne możemy przedstawić w postaci sumy trzech różnych liczb, których największy wspólny dzielnik (NWD) jest większy od 1. Taką liczbą jest np. liczba 12=2+4+6, gdzie NWD(2,4,6)=2, także wiele innych choć nie wszystkie (np. nie istnieje taka suma dla liczby 10). Co więcej dla wielu liczb można znaleźć więcej takich sum np. 22=2+4+16, ale także 4+8+10. W obu przypadkach NWD z elementów będących składnikami tych sum wynosi naturalnie 2.

Twoje zadanie polegać będzie na napisaniu programu, który dla pewnego ciągu liczb naturalnych na podstawie możliwych przedstawień liczby przy pomocy sumy trzech różnych liczb, których NWD jest większe niż 1 zakwalifikuje każdą z liczb tego ciągu do osobnych zbiorów wyznaczanych ze względu na to ile wynosiło NWD składników sumujących się do danej liczby. Zauważmy, że niekiedy liczba może być zakwalifikowana kilkakrotnie do różnych zbiorów np. liczba 75.

Ponieważ 75= 9+63+3, ale także 75=5+20+50 (to naturalnie nie jedyne dla tej liczby sumy spełniające warunki zadania) więc już z tego tytułu liczba 75 powinna się znaleźć w zbiorze, charakteryzowanym przez NWD, które dla składników sumujących się do niej jest równe 3, jak i w zbiorze charakteryzowanym przez NWD składników równe 5.

Dane wejściowe

Plik tekstowy ***ciag.txt*** zawierający w pierwszym wierszu liczbę naturalną **n** oznaczającą ilość liczb w ciągu (1<=n<=40). W kolejnych n wierszach znajdują się liczby naturalne (po jednej w wierszu, o których wiemy, że są większe od 1 oraz mniejsze od $10^{4)}$.

Dane wyjściowe

Zapisane w kolejnych wierszach pliku ***NWD3.txt*** : wartość NWD , a po dwukropku w odstępach między sobą kolejne liczby, które dały się przedstawić w postaci sumy trzech różnych liczb o tym właśnie NWD (nie ma znaczenia ile razy dała się tak przedstawić, wystarczy, ze istniał choć jeden układ trzech elementów składających się na daną liczbę, których NWD odpowiada temu z tego wiersza). Wiersze powinny być uporządkowane rosnąco ze względu na wartość NWD (naturalnie najniższa możliwa wartość NWD jest równa 2). W plik ***NWD3.txt*** nie umieszczamy w ogóle liczb, które nie dawały się przedstawić w postaci choćby jednej sumy trzech różnych składników, których NWD byłoby większe niż 1. Jeśli zaś żadna z liczb z pliku ***ciagi.txt*** nie da się w ten sposób przedstawić to w pliku ***NWD3.txt*** powinno być umieszczone jedynie słowo BRAK.

Przykład 1

Jeżeli w pliku ***ciag.txt*** znajdują się następujące dane:

5

12

4

75

22

15

to w pliku ***NWD3.txt*** powinna się znaleźć następujące wiersze:

2: 12 22

3: 75

5: 75

Wyjaśnienie: przedstawienia liczby 75 za pomocą sumy trzech różnych składników o NWD=3 oraz sumy trzech innych składników o NWD=5, a także przedstawienia dla liczb 12 i 22 (obie liczby są sumą trzech różnych składników o NWD=2) zostały pokazane w treści zadania, natomiast liczb 4 oraz 15 nie da się przedstawić w postaci sumy trzech różnych składników o NWD >1.

Do oceny oddajesz plik zawierający kod źródłowy programu o nazwie *Zadanie5.*

11 punktów