Zadanie 60.

Wiązka zadań *Dzielniki*

W pliku liczby.txtdanych jest 200 różnych liczb całkowitych z przedziału [2, 1 000 000], każda w osobnym wierszu pliku. Napisz program (lub kilka programów), który poda odpowiedzi do poniższych zadań. Odpowiedzi zapisz do pliku wyniki.txt.

60.1.

Policz, ile jest w pliku wejściowym liczb mniejszych niż 1000, oraz podaj dwie takie liczby, które pojawiają się w pliku jako ostatnie (możesz założyć, że będą co najmniej dwie).

60.2.

Wśród liczb występujących w pliku wejściowym znajdź te, które mają dokładnie 18 dzielników naturalnych (wliczając w nie 1 i samą liczbę). Dla każdej znalezionej liczby wypisz, oprócz jej wartości, listę wszystkich jej dzielników, posortowaną rosnąco.

60.3.

Znajdź największą liczbę w pliku, która jest względnie pierwsza ze wszystkimi pozostałymi, czyli taką, która z żadną z pozostałych liczb nie ma wspólnego dzielnika innego niż 1.

Zadanie 61.

Wiązka zadań *Ciągi arytmetyczne*

Ciąg liczb całkowitych nazywamy *ciągiem arytmetycznym*, jeśli różnica między każdymi dwoma kolejnymi jego wyrazami jest identyczna. Ciągiem arytmetycznym jest na przykład ciąg (1, 3, 5, 7, 9). Jest to ciąg o różnicy 2, ponieważ każdy wyraz tego ciągu, poza pierwszym, różni się od poprzedniego wyrazu o 2. Ciąg (17, 22, 27, 32, 37) jest ciągiem arytmetycznym o różnicy 5. W tym zadaniu rozpatrujemy tylko takie ciągi arytmetyczne, które mają dodatnią różnicę oraz co najmniej pięć wyrazów.

W pliku *ciagi.txt* danych jest 100 ciągów składających się z liczb całkowitych dodatnich, nieprzekraczających 1 000 000. Każdy ciąg opisany jest w dwóch wierszach: pierwszy zawiera liczbę wyrazów ciągu (co najmniej 5 i co najwyżej 1000), zaś drugi — kolejne wyrazy ciągu, oddzielone pojedynczymi odstępami. Dla przykładu pierwsze cztery wiersze pliku mają następującą postać:

5

1 3 6 7 9

5

17 22 27 32 37

Napisz program (lub kilka programów), który wykona podane poniżej polecenia.

61.1.

Podaj, ile spośród podanych w pliku ciagi.txt ciągów jest ciągami arytmetycznymi. Znajdź wśród nich ciąg o największej różnicy i oblicz jego różnicę. Liczbę ciągów arytmetycznych oraz największą różnicę zapisz w pliku wynik1.txt*.*

61.2.

Dla każdego podanego ciągu znajdź — jeśli istnieje — największą występującą w nim liczbę, która jest pełnym sześcianem jakiejś liczby naturalnej (w pierwszym z przykładowych ciągów jest to 1 = 13, w drugim — 27 = 33).

Znalezione liczby (po jednej dla każdego ciągu, w którym taka liczba występuje) zapisz w pliku wynik2.txt, w kolejności zgodnej z kolejnością ciągów, z których pochodzą.

61.3.

Plik bledne.txt ma identyczną strukturę jak ciagi.txt, ale zawiera tylko 20 ciągów. Wiadomo jednak, że wszystkie występujące w nim ciągi są ciągami arytmetycznymi z jednym błędem: jeden z wyrazów w każdym ciągu został zastąpiony przez liczbę naturalną nienależącą do ciągu.

Dla każdego ciągu znajdź i wypisz błędny wyraz. Odpowiedzi zapisz w pliku wynik3.txt, podając dla każdego ciągu błędną liczbę w osobnym wierszu, w kolejności zgodnej z kolejnością ciągów w pliku wejściowym.

Zadanie 62.

Wiązka zadań *Liczby ósemkowe*

W pliku liczby1.txt znajduje się 1000 liczb całkowitych dodatnich, zapisanych ósemkowo, maksymalnie sześciocyfrowych. Każda liczba umieszczona jest w osobnym wierszu.

W pliku liczby2.txt znajduje się 1000 liczb całkowitych dodatnich, zapisanych dziesiętnie, maksymalnie sześciocyfrowych. Każda liczba umieszczona jest w osobnym wierszu.

Napisz program(-y), za pomocą którego(-ych) rozwiążesz poniższe zadania. Do oceny oddaj dokument wyniki.txt z rozwiązaniami poszczególnych zadań oraz pliki źródłowe programów wykorzystanych do uzyskania rozwiązania.

62.1.

Wyszukaj w pliku liczby1.txt dwie liczby, najmniejszą i największą. Podaj wartości tych liczb w zapisie ósemkowym.

62.2.

Znajdź najdłuższy niemalejący ciąg liczb występujących w kolejnych wierszach pliku liczby2.txt. Podaj pierwszy element tego ciągu oraz liczbę jego elementów. Możesz założyć, że jest jeden taki ciąg.

Dla przykładowych danych:

23156

1231

1345

1456

1456

897

najdłuższy niemalejący ciąg liczb rozpoczyna się liczbą 1231 i składa się z 4 elementów.

62.3.

Porównaj wartości liczb zapisanych w wierszach o tych samych numerach w plikach liczby1.txt i liczby2.txt. Podaj liczbę wierszy, w których:

a) liczby mają w obu plikach taką samą wartość;

b) wartość liczby z pliku liczby1.txt jest większa od wartości liczby z pliku liczby2.txt.

Dla przykładowych danych:

liczby1.txt liczby2.txt

11456 1302

22666 9654

546 499

odp. a) 1 wiersz, bo tylko w drugim wierszu liczby mają taką samą wartość: 226668 = 965410

odp. b) 1 wiersz, bo tylko w pierwszym wierszu wartość liczby w pierwszym pliku jest większa niż odpowiadająca jej wartość w drugim pliku: 114568>130210.

62.4.

Podaj, ile razy w zapisie dziesiętnym wszystkich liczb z pliku liczby2.txt występuje cyfra 6 oraz ile razy wystąpiłaby ta cyfra, gdyby te same liczby były zapisane w systemie ósemkowym.

Zadanie 63.

Wiązka zadań *Ciągi zerojedynkowe*

W pliku ciagi.txt w oddzielnych wierszach znajduje się **1000** **różnych ciągów zerojedynkowych**, każdy o długości od 2 do 18. **Napisz program(-y)**, który pozwoli rozwiązać poniższe zadania. Następnie je rozwiąż, a odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym wyniki\_ciagi.txt. Wyniki do każdego zadania poprzedź numerem oznaczającym to zadanie.

63.1.

**Ciągiem dwucyklicznym** będziemy nazywać taki ciąg zerojedynkowy o długości parzystej, który składa się z dwóch fragmentów oraz , , takich że . Podaj wszystkie ciągi dwucykliczne zapisane w pliku ciagi.txt.

Przykład

Dla zestawu ciągów:

10010101010011001010101001

11001101001

1001000

11001100

101010011100

110011110011

3 podkreślone ciągi są dwucykliczne.

63.2.

Podaj liczbę ciągów z pliku ciagi.txt, w których nie występują obok siebie dwie jedynki.

Przykład

Dla zestawu ciągów:

10101010100101001010010101

11001101001

10001000

101010011100

000011

wynikiem jest liczba 2 (w dwóch podkreślonych ciągach dwie jedynki nie występują obok siebie).

63.3.

**Liczbą półpierwszą** nazywamy taką liczbę, która jest iloczynem dwóch liczb pierwszych. Podaj, ile ciągów z pliku ciagi.txt jest reprezentacją binarną liczb półpierwszych. Dodatkowo podaj największą i najmniejszą liczbę półpierwszą w zapisie dziesiętnym.

Przykład

Dla zestawu ciągów:

100010

1101001001

1100101

1111111111

10010110000010010010

podkreślone ciągi są zapisem binarnym liczb półpierwszych, ponieważ:

(100010)2 = 34= 2 \* 17, więc jest liczbą półpierwszą;

(1101001001)2 = 841= 29 \* 29, więc jest liczbą półpierwszą;

(1100101)2 = 101 = 101 \* 1;

(1111111111)2 = 1023 = 3 \* 11 \* 31;

(10010110000010010010)2 = 614546 = 2 \* 307273, więc jest liczbą półpierwszą.

Zadanie 64.

Wiązka zadań *Obrazki*

Bit parzystości ciągu złożonego z zer i jedynek jest równy 0, gdy w ciągu tym występuje parzysta liczba jedynek, w przeciwnym razie bit parzystości jest równy 1.

Czarno-biały obrazek rozmiaru *n* × *n* składa się z *n* wierszy po *n* pikseli. Każdy wiersz pikseli reprezentujemy jako ciąg zer i jedynek, każdy biały piksel reprezentujemy przez 0, czarny — przez 1. Na końcu każdego wiersza dodany jest bit parzystości, podobnie pod ostatnim wierszem obrazka dołączony jest wiersz bitów parzystości każdej z *n* kolumn. **Bitów parzystości nie traktujemy jako części obrazka.**

**Przykład**: Poniżej podajemy obrazek rozmiaru 5 × 5 oraz jego reprezentację, wraz z odpowiednimi bitami parzystości (bity parzystości zostały podkreślone):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | 0 1 1 0 1 1 |  |
|  |  |  |  |  |  | 1 1 1 0 1 0 |  |
|  |  |  |  |  |  | 1 1 1 1 1 1 |  |
|  |  |  |  |  |  | 0 1 1 0 0 0 |  |
|  |  |  |  |  |  | 1 1 0 1 1 0 |  |
|  |  |  |  |  |  | 1 1 0 0 0 |  |

Obrazek Reprezentacja

Plik dane\_obrazki.txt składa się z opisu 200 czarno-białych obrazków o rozmiarze 20 × 20 pikseli. **Sąsiednie obrazki oddzielone są w pliku pustym wierszem.**

Napisz program(-y), który poda odpowiedzi na pytania postawione w poniższych zadaniach. Odpowiedzi zapisz w plikuwyniki\_obrazki.txt. Odpowiedź do każdego zadania rozpocznij w nowym wierszu, poprzedzając ją numerem zadania.

64.1.

Obrazek nazywamy *rewersem*, jeśli liczba występujących w nim pikseli czarnych jest większa od liczby pikseli białych.

**Przykład:** W obrazku z powyższego przykładu występuje 18 pikseli czarnych i 7 pikseli białych. Zatem jest on rewersem.

Podaj, ile jest w pliku obrazków, które są rewersami. Podaj też największą liczbę pikseli czarnych występujących w jednym obrazku.

64.2.

Obrazek rozmiaru *n* × *n* będziemy nazywać *rekurencyjnym,* jeśli *n* jest parzyste oraz obrazek składa się z 4 kopii tego samego obrazka rozmiaru .

Przykład

Poniżej podajemy 3 obrazki rozmiaru 6 × 6, które są rekurencyjne.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Natomiast poniższe obrazki nie są rekurencyjne:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Podaj liczbę obrazków rekurencyjnych w pliku wejściowym. Ponadto podaj opis pierwszego obrazka rekurencyjnego występującego w pliku. W opisie obrazka pomiń bity parzystości (pamiętaj, że obrazek składa się z 20 wierszy po 20 pikseli, które reprezentujemy jako ciąg zer i jedynek).

64.3.

Obrazek nazywamy *poprawnym*, jeśli wszystkie bity parzystości są w nim poprawne (zarówno w wierszach, jak i kolumnach). Obrazek nazywamy *naprawialnym*, jeśli nie jest poprawny, a jednocześnie co najwyżej jeden bit parzystości wiersza i co najwyżej jeden bit parzystości kolumny jest w nim niepoprawny.

Natomiast *nienaprawialnym* nazywamy obrazek, który nie jest poprawny i nie jest naprawialny.

Przykład

Poniżej podajemy przykłady obrazków poprawnych, naprawialnych i nienaprawialnych rozmiaru 5 × 5. Niepoprawne bity parzystości w obrazkach zostały wyróżnione podkreśleniem.

011000 011011 011011 011011 011011

111010 11101**1** 11101**1** 11101**1** 11101**1**

111111 111111 111111 111111 111111

011000 011000011000 01100**1** 01100**1**

110011 110110 110110 110110 110110

11011 11**1**00 11000 11000 **0**1000

poprawny naprawialny naprawialny nienaprawialny nienaprawialny

Podaj liczbę obrazków poprawnych, liczbę obrazków naprawialnych oraz liczbę obrazków nienaprawialnych. Ponadto podaj największą liczbę błędnych bitów parzystości występujących w jednym obrazku.

64.4.

W obrazku naprawialnym wystarczy zmienić jedną wartość, aby uzyskać obrazek poprawny. Dokładniej, jeśli niepoprawne są bity parzystości *i*-tego wiersza i *j*-tej kolumny, wystarczy zmienić *j*-ty piksel w *i*-tym wierszu. Jeśli niepoprawny jest dokładnie jeden bit parzystości (wiersza **albo** kolumny), wystarczy zmienić ten bit parzystości.

Przykład

Rozważmy następujące dwa obrazki naprawialne rozmiaru 5 × 5 (niepoprawne bity parzystości w obrazkach zostały podkreślone).

Obraz 1: Obraz 2:

011011 011011

11**1**01**1** 11101**1**

111111 111111

011000 011000

110110 110110

11**1**00 11000

Zmieniając trzecią jedynkę w drugim wierszu pierwszego obrazka, uzyskamy obrazek poprawny (niepoprawne były bity parzystości trzeciej kolumny i drugiego wiersza). Zmieniając niepoprawny bit parzystości w drugim wierszu drugiego obrazka z 1 na 0, również uzyskamy obrazek poprawny.

Podaj numery obrazków naprawialnych, przyjmując, że numery kolejnych obrazków w pliku to 1, 2, 3 itd. Przy numerze każdego obrazka naprawialnego podaj numer wiersza i kolumny wartości, którą wystarczy zmienić, aby uzyskać obrazek poprawny.

Zadanie 65.

Wiązka zadań *Ułamki*

W pliku dane\_ulamki.txt znajduje się 1000 par liczb naturalnych dodatnich, mniejszych niż 12 000. Każda para liczb jest zapisana w osobnym wierszu, liczby w wierszu rozdzielone są pojedynczym znakiem odstępu. Parę liczb zapisanych w tym samym wierszu interpretujemy jako ułamek, którego licznikiem jest pierwsza liczba, a mianownikiem — druga liczba.

Przykład

Plik o zawartości

3 6

2 3

5 3

2 4

15 5

odpowiada ułamkom  .

*Postacią* *nieskracalną* ułamka  nazywamy taki ułamek , że = oraz *x* i *y* są względnie pierwsze (czyli *x* i *y* nie mają wspólnego dzielnika większego od 1).

**Napisz program(-y)**, który poda odpowiedzi na pytania postawione w poniższych zadaniach. Odpowiedzi zapisz w plikuwyniki\_ulamki.txt. Odpowiedź do każdego zadania podaj w osobnym wierszu, poprzedzając ją numerem zadania.

65.1.

Podaj ułamek o minimalnej wartości. Jeśli w pliku występuje więcej niż jeden taki ułamek, to podaj ten spośród nich, który ma najmniejszy mianownik. Twoja odpowiedź powinna zawierać parę liczb oznaczającą licznik i mianownik ułamka.

Przykład

Dla podanego powyżej pliku, opisującego ułamki , minimalną wartość mają ułamki . Ponieważ  ma mniejszy mianownik niż , więc odpowiedzią jest para liczb: 2 i 4.

65.2.

Podaj liczbę zapisanych w pliku ułamków, które zostały podane w postaci nieskracalnej.

Przykład

Dla podanego powyżej pliku, opisującego ułamki , w postaci nieskracalnej zapisane zostały . Natomiast i nie są ułamkami w postaci nieskracalnej (ich liczniki i mianowniki dzielą się odpowiednio przez 3 i 2), podobnie  (jego licznik i mianownik dzielą się przez 5). Zatem odpowiedzią jest liczba 2.

65.3.

Zapis danych w postaci nieskracalnej uzyskamy, zamieniając każdy ułamek na jego postać **nieskracalną**. Podaj sumę liczników wszystkich podanych w pliku ułamków, jaką otrzymalibyśmy po sprowadzeniu ułamków do nieskracalnej postaci.

Przykład

Dla podanego powyżej pliku, opisującego ułamki , nieskracalne postacie kolejnych ułamków to: . Suma liczników tych ułamków to 1+2+5+1+3=12. Zatem odpowiedzią jest 12.

65.4.

Ułamki w pliku zostały tak dobrane, że każdy mianownik jest dzielnikiem liczby b=22⋅32⋅52⋅72⋅13, a wartość każdego ułamka jest nie większa niż 3. Oznacza to, że sumę wszystkich ułamków można przedstawić jako ułamek , którego mianownikiem jest b=22⋅32⋅52⋅72⋅13. Wyznacz sumę ułamków ze wszystkich wierszy i podaj licznik takiego ułamka, że suma ułamków jest równa .

Przykład

Dla podanego powyżej pliku, opisującego ułamki 1/2, 2/3, 5/3, 2/4, 15/5, suma ułamków to:

,

gdzie b=22⋅32⋅52⋅72⋅13. Poprawna odpowiedź wynosi więc .

Zadanie 66.

Wiązka zadań *Trójki liczb*

W pliku trojki.txt w oddzielnych wierszach znajduje się 1000trójek liczb naturalnych z przedziału od 1 do 550000000. W każdym wierszu są umieszczone trzy liczby rozdzielone pojedynczymi odstępami.

Przykład

3 4 5

12 5 13

12 491 17

11 13 143

15 28 91

**Napisz program(-y)**, który da odpowiedzi do poniższych zadań. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym wyniki\_trojki.txt. Wyniki do każdego zadania poprzedź numerem oznaczającym to zadanie.

66.1.

Wypisz wszystkie trójki liczb z pliku trojki.txt, w których suma cyfr dwóch pierwszych liczb jest równa ostatniej (trzeciej) liczbie.

Przykład

12 491 17

1+2+4+9+1=17

66.2.

Wypisz wszystkie wiersze z pliku trojki.txt zawierające takie trzy liczby *a*, *b*, *c*, w których *a* i *b* są liczbami pierwszymi oraz *c* = *a* ∙ *b*.

Przykład

11 13 143

11 i 13 są liczbami pierwszymi i 11 ∙ 13 = 143

66.3.

Wypisz z pliku trojki.txt wszystkie pary sąsiadujących ze sobą wierszy, takie że liczby w tych wierszach są długościami boków trójkątów prostokątnych.

Przykład

3 4 5 32+42=52

12 5 13 52+122=132

66.4.

Podaj, ile jest w pliku trojki.txt wierszy, w których znajdują się liczby reprezentujące długości boków trójkąta. Ciąg wierszy nazywamy **trójkątnym**, jeśli liczby w każdym wierszu reprezentują długości boków trójkąta. Podaj długość najdłuższego ciągu trójkątnego w pliku.

Zadanie 67.

Wiązka zadań *Binarny fraktal Fibonacciego*

Ciąg Fibonacciego to [ciąg](http://pl.wikipedia.org/wiki/Ci%C4%85g_(matematyka)) [liczb naturalnych](http://pl.wikipedia.org/wiki/Liczby_naturalne) określony [rekurencyjnie](http://pl.wikipedia.org/wiki/Rekurencja) w sposób następujący:

a każdy następny element ciągu jest sumą dwóch poprzednich, czyli:

Binarny fraktal Fibonacciego to dwuwymiarowa tablica zawierająca w kolejnych wierszach binarne zapisy kolejnych liczb Fibonacciego, gdzie każde zero w zapisie zastąpiono białym kwadratem, a każdą jedynkę czarnym kwadratem (p. rysunek). Wszystkie binarne zapisy powinny składać się z jednakowej liczby cyfr, czyli do zapisów krótszych niż najdłuższy należy dodać zera wiodące.

Przykład binarnego fraktala dla pierwszych 10 liczb Fibonacciego:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *n* | *Fn* | zapis binarny *Fn* | Binarny fraktal Fibonacciego | | | | | | | |
| 1 | 1 | 000001 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 1 | 000001 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3 | 2 | 000010 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | 3 | 000011 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 5 | 5 | 000101 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 6 | 8 | 001000 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 7 | 13 | 001101 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 8 | 21 | 010101 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 9 | 34 | 100010 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 10 | 55 | 110111 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |

Napisz program komputerowy, za pomocą którego uzyskasz odpowiedzi do poniższych zadań. Rysunek fraktala (zadanie nr 3) wykonaj, wykorzystując dostępne narzędzia informatyczne. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym o nazwie wyniki.txt, natomiast rysunek fraktala w pliku fraktal.xxx, gdzie xxx oznacza rozszerzenie pliku, w którym zapisany jest obraz fraktala.

67.1.

Podaj wartości *F*10, *F*20, *F*30, *F*40. Zapisz każdą z liczb w osobnym wierszu.

67.2.

Znajdź wszystkie liczby pierwsze wśród liczb *F*1, *F*2, … , *F*40. Zapisz każdą z liczb w osobnym wierszu.

67.3.

Dla pierwszych 40 liczb Fibonacciego utwórz binarny fraktal Fibonacciego:

* Wypisz reprezentację binarną wszystkich liczb Fibonacciego od *F*1 do *F*40.
* Wyrównaj długości reprezentacji binarnych wszystkich liczb Fibonacciego od *F*1 do *F*40 i na ich podstawie sporządź obraz binarnego fraktala Fibonacciego.

67.4.

Podaj w zapisie binarnym wyrazy ciągu Fibonacciego z zakresu od *F*1 do *F*40, które w tym zapisie mają dokładnie 6 jedynek.

Zadanie 68.

Wiązka zadań *Napisy — anagramy*

Dwa napisy *a* i *b* są swoimi **anagramami**, jeżeli napis *a* (napis *b*) można zapisać za pomocą liter występujących w napisie *b* (napisie *a*), wykorzystując **wszystkie** jego litery.

W pliku dane\_napisy.txt znajduje się 1000 par napisów, z których każdy jest długości od 2 do 20 znaków, składających się z wielkich liter: A, B, C, D, E, F, G, H, I, J. Każda para napisów jest zapisana w osobnym wierszu, a napisy oddzielone są pojedynczym znakiem odstępu.

Przykład

AIHAHGHBEAFJAJDI HGIHFEHHJGBCBGD

FBJHCFFGADD EHADJAJBJBEGD

JHGHADJ AGFEHHEHIAEJFC

EJJHFHIGCEBDAIB DCAFFDICGBEAHAEJ

FBAEEGICHFFACICIGB EEHAHHCABHDHGDFFEGD

**Napisz program(-y)**, który poda odpowiedzi dla następujących zadań. Odpowiedzi zapisz w plikuwyniki\_anagramy.txt. Odpowiedź do każdego zadania podaj w osobnym wierszu, poprzedzając ją identyfikatorem zadania.

68.1.

Napis nazywamy **jednolitym**, jeżeli wszystkie jego litery są takie same. Przykładem takiego napisu jest *AAAA*. Podaj liczbę wierszy zawierających parę napisów jednolitych, które są wzajemnie swoimi anagramami.

Przykład

Dla pliku zawierającego następujące dane:

*AAAA* *AAAA*

*AHHAH AHHAH*

*AAAA AAAAAAA*

*BBBBBBB BABBAB*

*CCCCC CCCCC*

wynikiem jest liczba 2 (pierwszy i ostatni wiersz). Zwróć uwagę, że napisy w trzecim wierszu są napisami jednolitymi, ale nie są wzajemnie swoimi anagramami.

68.2.

Podaj liczbę wierszy, które zawierają napisy będące wzajemnie swoimi **anagramami**.

Przykład

Dla pliku zawierającego następujące dane:

*BBBAAB BBBABA*

*AAAA* *AAAAA*

*AHHAH AHHAH*

*BBABBABB BBBABB*

*BABABB CACACC*

wnikiem jest liczba 2 (pierwszy i trzeci wiersz).

68.3.

Podaj największą liczbę *k* taką, że w pliku znajduje się *k* napisów, z których każde dwa są wzajemnie swoimi anagramami.

Przykład

Dla pliku zawierającego następujące dane:

*BABABB BBBABA*

*AAAA* *AAAA*

*AHHAH AHHAH*

*BBABBABB BABBAB*

*BBAABB CCCCC*

wnikiem jest liczba 4 (*BABABB BBBABA BABBAB BBAABB*).

Zadanie 69.

Wiązka zadań *Geny*

Informację genetyczną (genotyp) każdego osobnika z galaktyki Madgen opisuje słowo (napis), w którym występują litery ze zbioru {A, B, C, D, E}. Obowiązują przy tym następujące zasady:

1. Organizmy żyjące na Madgen tworzą gatunki *g1, g2, g3,* …, gdzie *gi* to zbiór osobników o długości genotypu równej *i*.
2. W skład genotypu mogą wchodzić geny. Pierwszy gen rozpoczyna się pierwszą występującą w genotypie sekwencją AA, a kończy się najbliższą napotkaną po niej sekwencją BB. Każdy kolejny gen rozpoczyna się pierwszą sekwencją AA, występującą za końcem poprzedniego genu, i analogicznie kończy się najbliższą napotkaną sekwencją BB.
3. Geny nazywamy częścią kodującą genotypu, pozostałe fragmenty tworzą część niekodującą.

Przykład 1.

Genotyp AACDBABBBCDAABCBBAAE

zawiera geny AACDBABB oraz AABCBB. Zwróćmy uwagę, że:

* ciąg AA występujący za genem AABCBB nie jest początkiem genu, ponieważ nie występuje za nim ciąg BB kończący gen;
* część kodująca genotypu AACDBABBBCD***AABCBB***AAE jest równa  
  AACDBABB***AABCBB***.

Przykład 2.

Genotyp AADBAADDDDEEEBBEE

zawiera gen **AA**DB**AA**DDDDEEEBB. Zwróćmy uwagę, że:

* pierwsze pojawienie się ciągu AA determinuje początek genu, dlatego w powyższym genotypie występuje gen AADBAADDDDEEEBB, a nie gen AADDDDEEEBB.

Plik dane\_gen.txt zawiera genotypy 1000 osobników z galaktyki Madgen. Każdy wiersz pliku zawiera genotyp jednego osobnika o długości nie większej niż 500 znaków.

Przykład

ABAEACBAADAACAABBABCDA

ABAEACBADEADACACABBABCDA

Napisz program(-y), który poda odpowiedzi na pytania postawione w poniższych zadaniach. Odpowiedzi zapisz w plikuwyniki\_gen.txt. Odpowiedź do każdego zadania rozpocznij w nowym wierszu, poprzedzając ją numerem zadania.

69.1.

Podaj liczbę wszystkich gatunków, których genotypy zapisane są w pliku dane\_gen.txt. Podaj największą liczbę osobników reprezentujących ten sam gatunek.

69.2.

Występowanie w jakimkolwiek **genie** ciągu **BCDDC** oznacza mutację powodującą małą odporność osobnika na zmęczenie. Podaj, ile osobników spośród tych, których genotypy zapisane są w pliku, ma tę mutacje.

Przykład

Osobnik o genie AAC**BCDDC**BBACDE ma mutację BCDDC (ciąg BCDDC występuje w obrębie podkreślonego genu), natomiast osobnik o genie C**BCDDC**BBAACDEBB nie ma tej mutacji, gdyż występujący ciąg BCDDC nie jest ulokowany w żadnym genie.

69.3.

Wyznacz i podaj największą liczbę genów występujących u jednego osobnika. Podaj też największą długość genu zapisanego w całym pliku.

Przykład

Rozważmy plik składający się z genotypów:

EAABCDBBDCBAAE

EAABCDBBDCBAAEBCDBBEE

EAABCDBBECAAB

Pierwszy osobnik ma jeden gen (AABCDBB), drugi ma dwa geny (AABCDBB i AAEBCDBB), a trzeci osobnik ma jeden gen (AABCDBB). Zatem największa liczba genów u jednego osobnika wynosi 2, a największa długość genu to 8 (gen AAEBCDBB ma tę długość).

69.4.

Genotyp odczytywany z materiału biologicznego może być odkodowany w kierunku od strony lewej do prawej lub odwrotnie: od strony prawej do lewej. Genotyp nazywać będziemy *odpornym*, jeśli czytany od strony lewej do prawej oraz od strony prawej do lewej ma dokładnie taką samą część kodującą. Natomiast genotyp *silnie odporny* to taki, który czytany od strony lewej do prawej oraz od strony prawej do lewej daje dokładnie ten sam napis. (Inaczej mówiąc, genotyp jest silnie odporny, gdy jest palindromem).

Przykład

Rozważmy genotypy:

EAABCDBBDCBAAE

EAABCDBBDCBAAEBCDEE

EAABCDBBECAAB

Genotyp EAABCDBBDCBAAE jest silnie odporny (jest palindromem). Genotyp EAABCDBBDCBAAEBCDEE nie jest silnie odporny (nie jest palindromem), ale jest odporny, gdyż czytany od strony lewej do prawej, jak i od strony prawej do lewej ma taką samą część kodującą: AABCDBB. Natomiast genotyp EAABCDBBECAAB nie jest silnie odporny (nie jest palindromem), nie jest też odporny, gdyż czytany od strony lewej do prawej daje część kodującą AABCDBB, a czytany od strony prawej do lewej ma część kodującą równą AACEBB.

Wyznacz liczbę genotypów odpornych oraz liczbę genotypów silnie odpornych.

Zadanie 70.

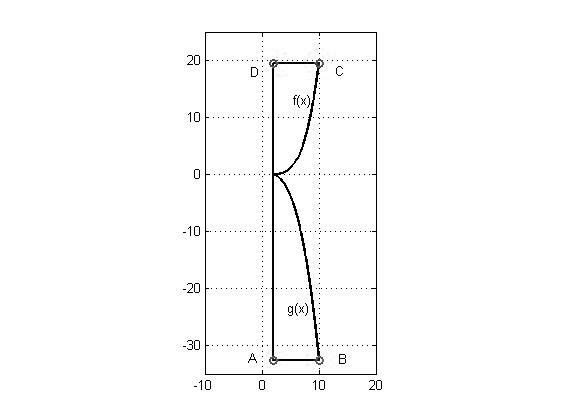
Wiązka zadań *Zasłona*

Pani Binarna dostała zlecenie na uszycie zasłony. Na rysunku poniżej przedstawiono zasłonę, która jest ograniczona:

* od góry prostą ,
* od dołu prostą ,
* z lewej strony prostą ,
* z prawej strony dwoma krzywymi:  oraz .

**Uwaga:** Zauważ, że , zaś .

Rysunek pomocniczy:



Korzystając z dostępnych narzędzi informatycznych, wykonaj poniższe zadania. Odpowiedzi do nich umieść w pliku zadanie\_zaslona.txt. Każda odpowiedź powinna być poprzedzona numerem je oznaczającym.

70.1.

Pani Binarna zakupiła tyle materiału, ile wynosi pole prostokąta ABCD, w którym mieści się zasłona. Oblicz, jaka będzie powierzchnia materiału pozostałego po wykrojeniu zasłony. Wynik podaj z dokładnością do 1/1000.

70.2.

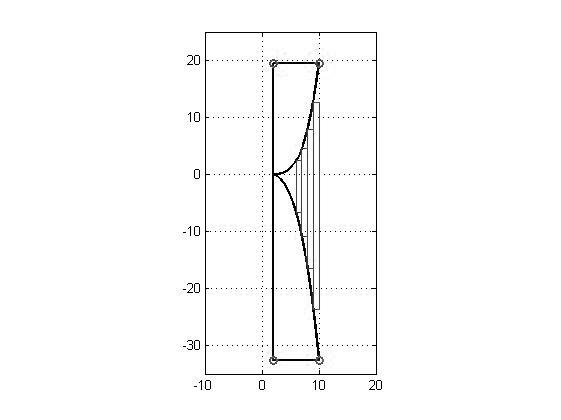
Pani Binarna zamierza obszyć taśmą zasłonę ze wszystkich czterech stron, w tym celu chce wyznaczyć obwód zasłony. Część obwodu ograniczoną wykresem funkcji *f*(*x*) szacujemy w następujący sposób: Odcinek [2,10] dzielimy na 1000 równych części, których prawe końce oznaczamy przez *x*1,…,*x*1000. Długość krzywej odpowiadającej wykresowi *f*(*x*) na przedziale [2,10] przybliżamy długością łamanej łączącej punkty (2, *f*(2)), (*x*1, *f*(*x*1)), (*x*2, *f*(*x*2)) itd. aż do (*x*1000, *f*(*x*1000)). Analogicznie wyznaczamy część obwodu ograniczoną przez *g*(*x*).

Stosując powyższą metodę wyznaczania obwodu, oblicz długość taśmy, jaką musi zakupić pani Binarna, zakładając, że w sprzedaży jest tylko taśma o długościach będących wielokrotnością jednego metra.

70.3.

Pani Binarna postanowiła wykorzystać pozostały fragment materiału i wyciąć z niego pasy o szerokości 0,25 m i o bokach równoległych do osi układu współrzędnych. Podaj sumę długości pasów, które można wyciąć z pozostałego fragmentu materiału. Załóż, że długość każdego wyciętego pasa jest liczbą całkowitą oraz że pani Binarna zaczyna wycinać pasy od prawej strony materiału.

Rysunek pomocniczy:

****

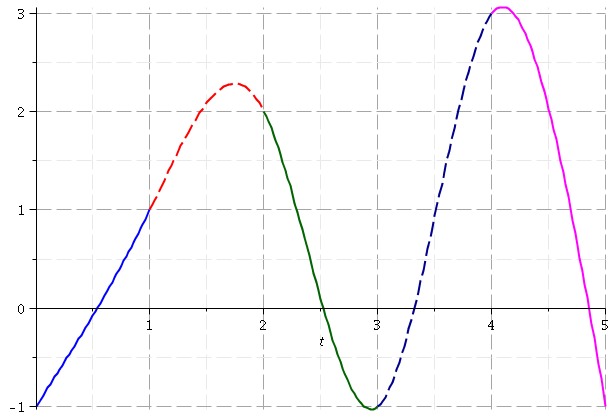
Zadanie 71.

Wiązka zadań *Funkcja*

Wykres funkcji *f* złożony jest z pięciu fragmentów:

gdzie każda z funkcji jest wielomianem stopnia trzeciego. W pliku funkcja.txt zapisane są współczynniki postaci ogólnej wielomianów ; w *i*-tym wierszu pliku zapisane są cztery liczby rzeczywiste: (oddzielone pojedynczym odstępem), dla których

Poniższy rysunek przedstawia wykres funkcji .



Napisz program, który da odpowiedzi do poniższych zadań. Zapisz je w pliku tekstowym wyniki\_funkcja.txt. Wyniki do każdego zadania poprzedź numerem je oznaczającym. Wszystkie wyniki należy wypisać, stosując zaokrąglenie do podanej liczby cyfr po przecinku.

71.1.

Podaj wartość z dokładnością do 5 cyfr po przecinku.

71.2.

Znajdź wartość , dla której wartość jest największa.

Jako wynik podaj wartość z dokładnością do trzech, a wartość z dokładnością do 5 cyfr po przecinku.

71.3.

Znajdź wszystkie miejsca zerowe funkcji w przedziale . Odpowiedzi podaj z dokładnością do 5 cyfr po przecinku.

Zadanie 72.

Wiązka zadań *Podobne napisy*

W pliku napisy.txt znajduje się 200 wierszy, z których każdy zawiera dwa napisy o długości od 1 do 50 znaków, oddzielone pojedynczym odstępem. Napisy składają się wyłącznie z małych liter alfabetu angielskiego.

Napisz program (lub kilka programów), który pozwoli rozwiązać poniższe zadania. Odpowiedzi zapisz w pliku wyniki.txt.

72.1.

Oblicz, w ilu wierszach jeden (którykolwiek) z napisów jest przynajmniej trzy razy dłuższy od drugiego. Jako odpowiedź wypisz liczbę takich wierszy oraz parę napisów z pierwszego z nich.

72.2.

Znajdź (i wypisz) wszystkie takie wiersze pliku, w których drugi napis da się otrzymać z pierwszego przez dopisanie na jego końcu pewnej dodatniej liczby liter (na przykład *kot* i *kotara*). Dla każdego wiersza podaj oba znajdujące się w nim napisy, a osobno wypisz litery, które należy dopisać.

72.3.

Niektóre z podanych par napisów mają identyczne zakończenia (na przykład *komputer* i *krater*). Znajdź i wypisz największą możliwą długość takiego zakończenia, a także wszystkie pary napisów w wierszach, które osiągają tę maksymalną długość.

Zadanie 73.

Wiązka zadań *Statystyki tekstu*

W pliku tekst.txtdany jest tekst złożony ze słów pisanych wielkimi literami alfabetu angielskiego. Słów jest 1876, oddzielone są one pojedynczymi odstępami, a inne znaki poza literami i spacjami w tekście nie występują. Napisz program(-y), który poda odpowiedzi do poniższych zadań. Odpowiedzi zapisz w pliku wyniki.txt.

73.1.

Oblicz, ile jest w tekście słów, w których występują dwie kolejne takie same litery.

73.2.

Sporządź statystykę częstotliwości występowania liter w tekście: dla każdej litery podaj liczbę jej wystąpień we wszystkich słowach tekstu oraz jej procentowy udział wśród wystąpień wszystkich liter w tekście(do statystyki nie wliczaj spacji). Odpowiedź zapisz w następującej postaci:

A: 632 (7.56%)

B: 196 (2.34%)

...

Wartości procentowe podaj w zaokrągleniu do dwóch miejsc po przecinku.

73.3.

Wśród słów w tekście policz długość najdłuższego podsłowa (fragmentu złożonego z kolejnych liter) złożonego z samych spółgłosek. Pamiętaj, że samogłoski to: A, E, I, O, U oraz Y, zaś pozostałe litery są spółgłoskami.

Podaj długość najdłuższego takiego podsłowa, liczbę słów, które zawierają podsłowo o takiej długości, oraz pierwsze z nich, które występuje w pliku tekst.txt**.**

Zadanie 74.

Wiązka zadań *Hasła*

W pliku hasla.txt danych jest 200 haseł użytkowników pewnego systemu. Każdy użytkownik posiada jedno hasło (każde zapisane jest w osobnym wierszu), które zawiera od 1 do 20 znaków alfanumerycznych, tzn. cyfr od 0 do 9 lub liter alfabetu łacińskiego (małych lub dużych). Polityka bezpieczeństwa systemu wymaga, aby hasła były odpowiednio skomplikowane i nie powtarzały się.

Poniżej podano pierwsze pięć haseł zapisanych w pliku hasla.txt:

ZXUhkPLcjKo

ikfLDegQXj

8Y7JGYXXR5

603624722555

50q4252ax5

Napisz program, który da odpowiedzi do poniższych zadań. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym wyniki\_hasla.txt. Wyniki do każdego zadania poprzedź numerem oznaczającym to zadanie.

74.1.

Podaj liczbę haseł złożonych jedynie ze znaków numerycznych, tzn. cyfr od 0 do 9.

74.2.

Wypisz hasła, które zostały użyte przez co najmniej dwóch różnych użytkowników, tzn. występujące w dwóch różnych wierszach. Hasła wypisz (bez powtórzeń) w kolejności leksykograficznej.

74.3.

Podaj liczbę użytkowników posiadających hasła, w których występuje fragment złożony z czterech kolejnych znaków ASCII (w dowolnej kolejności).

Przykłady haseł zawierających taki fragment to:

A5mnpoR89cd

A5876RRcg

As45FGHFEk90nba

75.4.

Podaj liczbę haseł, które spełniają jednocześnie poniższe warunki:

* hasło zawiera co najmniej jeden znak numeryczny, tzn. cyfrę od 0 do 9,
* hasło zawiera co najmniej jedną małą literę,
* hasło zawiera co najmniej jedną dużą literę.

Zadanie 75.

Wiązka zadań *Szyfr afiniczny*

Dany jest tekst złożony ze słów zbudowanych z małych liter alfabetu angielskiego. Metoda szyfrowania afinicznego — dla której *kluczem szyfrującym* są dwie liczby całkowite A i B — polega na wykonaniu kolejno następujących operacji:

* zastąpienia kolejnych liter alfabetu liczbami od 0 do 25: 'a' przez 0, 'b' przez 1, 'c' przez 2 itd. według przyporządkowania przedstawionego w poniższej tabeli:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

* pomnożenia liczby odpowiadającej każdej literze przez A i dodania otrzymanego wyniku do B,
* zamiany otrzymanych liczb z powrotem na litery; jeśli liczba jest większa niż 25, bierze się jej resztę z dzielenia przez 26.

Parametry klucza, czyli liczby A i B, powinny być liczbami całkowitymi z przedziału [0, 25].

Dla przykładu, jeśli kluczem szyfrującym jest (3, 7), czyli A = 3, zaś B = 7, to litera 'n' jest najpierw zastępowana liczbą 13. Po pomnożeniu jej przez A i dodaniu B otrzymujemy wynik równy 46. W następnym kroku otrzymujemy literę o numerze 46 – 26 = 20, czyli 'u'.

Okazuje się, że do odszyfrowania szyfru afinicznego można zastosować tę samą metodę, być może z innym kluczem. Na przykład, jeśli napis zaszyfrujemy kluczem (3, 7), to aby go odszyfrować, stosujemy ten sam algorytm z kluczem (9, 15). Dla przykładu, deszyfrując literę 'u' z kluczem (9, 15), otrzymamy liczbę 20 \* 9 + 15 = 195, czyli literę 'n', jako że 195 mod 26 = 13. Klucz (9,15) jest wówczas *kluczem deszyfrującym* dla klucza (3,7).

Napisz program(y), który poda odpowiedzi do poniższych zadań. Odpowiedzi zapisz do pliku wyniki.txt.

75.1.

W pliku tekst.txt dany jest, w pojedynczym wierszu, tekst złożony z dokładnie 805 słów zapisanych małymi literami alfabetu angielskiego, oddzielonych znakami odstępu. Żadne słowo nie jest dłuższe niż 15 znaków.

Znajdź i wypisz te słowa, których zarówno pierwszą, jak i ostatnia literą jest 'd'.

75.2.

Zaszyfruj szyfrem afinicznym o kluczu (5, 2) te słowa z pliku tekst.txt, które składają się z co najmniej 10 liter. Wypisz je w postaci zaszyfrowanej, po jednym w wierszu.

75.3.

Plik probka.txtskłada się z 5 wierszy, każdego zawierającego dwa napisy. Pierwszy z nich to pewne słowo zapisane tekstem jawnym, drugi zaś to to samo słowo zaszyfrowane za pomocą szyfru afinicznego (każde słowo innym kluczem).

Dla każdego z tych słów znajdź i wypisz klucz szyfrujący oraz klucz deszyfrujący.

Zadanie 76.

Wiązka zadań *Szyfr*

Rozważamy szyfrowanie przestawieniowe, w którym kluczem jest *n-elementowa* tablica zawierająca różne liczby całkowite z przedziału [1, *n*]. Na przykład kluczem *5*-elementowym może być tablica [3, 2, 5, 4, 1].

Szyfrowanie napisu *A* (o długości co najmniej *n*) kluczem *n*-elementowym *P*[1..*n*]odbywa się w następujący sposób:

* pierwsza litera słowa *A* zamieniana jest miejscami z literą na pozycji *P*[1],
* następnie druga litera słowa *A* zamieniana jest z literą na pozycji *P*[2]
* itd.

Uzyskane na końcu słowo jest szyfrem napisu *A* z kluczem *P*.

Jeśli napis *A* ma więcej niż *n* liter, to po *n*-tym kroku powyższego algorytmu kolejną literę zamieniamy znów z literą na pozycji *P*[1] itd. Oznacza to, że w *i*-tym kroku zamieniamy litery na pozycjach *i* oraz *P*[1+(*i*-1) mod *n*].

Przykład

Poniższa tabelka ilustruje szyfrowanie słowa „INFORMATYKA” kluczem P równym [3, 2, 5, 4, 1]:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| P[1+(i-1) mod n] | 3 | 2 | 5 | 4 | 1 | 3 | 2 | 5 | 4 | 1 | 3 |
| Słowo | I | N | F | O | R | M | A | T | Y | K | A |
| Krok 1 | **F** | N | **I** | O | R | M | A | T | Y | K | A |
| Krok 2 | F | **N** | I | O | R | M | A | T | Y | K | A |
| Krok 3 | F | N | **R** | O | **I** | M | A | T | Y | K | A |
| Krok 4 | F | N | R | **O** | I | M | A | T | Y | K | A |
| Krok 5 | **I** | N | R | O | **F** | M | A | T | Y | K | A |
| Krok 6 | I | N | **M** | O | F | **R** | A | T | Y | K | A |
| Krok 7 | I | **A** | M | O | F | R | **N** | T | Y | K | A |
| Krok 8 | I | A | M | O | **T** | R | N | **F** | Y | K | A |
| Krok 9 | I | A | M | **Y** | T | R | N | F | **O** | K | A |
| Krok 10 | **K** | A | M | Y | T | R | N | F | O | **I** | A |
| Krok 11 | K | A | **A** | Y | T | R | N | F | O | I | **M** |

Napis „KAAYTRNFOIM” jest zatem szyfrem napisu „informatyka” z kluczem [3, 2, 5, 4, 1]. **Napisz program**(-y), który da odpowiedzi do poniższych zadań.

76.1.

W pliku szyfr1.txt dane są:

* w wierszach o numerach od 1 do 6 — napisy złożone z 50 liter alfabetu łacińskiego;
* w wierszu nr 7 — klucz 50-elementowy; liczby oddzielone są pojedynczym odstępem.

Zaszyfruj wszystkie sześć napisów zgodnie z opisaną metodą. Wynik, czyli zaszyfrowane napisy, zapisz w osobnych wierszach w pliku wyniki\_szyfr1.txt.

76.2.

W pliku szyfr2.txt dane są:

* w pierwszym wierszu — napis złożony z 50 liter alfabetu łacińskiego;
* w drugim wierszu — klucz 15-elementowy; liczby oddzielone są pojedynczym odstępem.

Zaszyfruj dany napis zgodnie z opisaną metodą. Wynik, czyli zaszyfrowany napis, zapisz w pliku wyniki\_szyfr2.txt.

76.3.

W pliku szyfr3.txt dany jest napis złożony z 50 liter alfabetu łacińskiego. Napis ten powstał po zaszyfrowaniu pewnego napisu *A* kluczem [6, 2, 4, 1, 5, 3].

Podaj napis *A.* Wynik zapisz w pliku wyniki\_szyfr3.txt.

Zadanie 77.

Wiązka zadań *Szyfr Vigenère’a*

W zadaniu rozważamy teksty zbudowane tylko z wielkich liter alfabetu angielskiego, znaków odstępu i znaków przestankowych (przecinek, kropka). Oto litery alfabetu i numery ich pozycji w alfabecie:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

Szyfrowanie Vigenère’a polega na zastąpieniu każdej litery tekstu źródłowego literą odległą od niej cyklicznie w alfabecie o *k* pozycji.

Wartość *k* nie jest z góry ustalona dla całego tekstu źródłowego, lecz dla każdej litery w tekście jest określana osobno, w oparciu o słowo przyjęte jako klucz szyfrowania.

Przystępując do szyfrowania, należy przyporządkować kolejnym literom tekstu źródłowego kolejne litery klucza, chodząc po nim cyklicznie, jeśli jest krótszy od szyfrowanego tekstu. Znaki inne niż litery nie są szyfrowane, pomijamy je podczas przypisywania liter klucza. P**ozycja litery klucza** w alfabecie jest tą wartością ***k***, o jaką należy wykonać przesunięcie względem litery tekstu źródłowego w celu znalezienia odpowiadającej jej litery szyfru.

Przykład

tekst źródłowy: ”JEST OK”, klucz: ”EWA”

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| tekst źródłowy | J | E | S | T | spacja | O | K |
| Klucz | E | W | A | E |  | W | A |
| pozycja litery klucza | 4 | 22 | 0 | 4 |  | 22 | 0 |
| Szyfr | J→4 = **N** | E→22=**A** | S→0 = **S** | T→4 = **X** | spacja | O→22=**K** | K→0 = **K** |

wynik szyfrowania: ”NASX KK”.

Napisz w wybranym języku programowania program, który wyznaczy rozwiązania zadań podanych niżej. Wszystkie wyniki zapisz w pliku tekstowym Vigenere\_wyniki.txt, wyraźnie oddzielając odpowiedzi do poszczególnych zadań. Do oceny oddaj ten plik oraz plik (pliki) zawierający reprezentację komputerową rozwiązania.

77.1.

W pliku dokad.txt znajduje się jeden wiersz z tekstem. Długość tekstu nie przekracza 1024 znaków. Należy zaszyfrować ten tekst metodą Vigenère’a, używając jako klucza słowa: ”LUBIMYCZYTAC”.

1. Podaj liczbę powtórzeń klucza niezbędną do zaszyfrowania całego tekstu źródłowego (uwzględniając w nich ostatnie rozpoczęte powtórzenie).
2. Podaj zaszyfrowany tekst i zapisz go w pliku z odpowiedziami.

77.2.

W pliku szyfr.txt zapisano dwa wiersze. W pierwszym wierszu znajduje się tekst zaszyfrowany metodą Vigenère’a. W drugim wierszu znajduje się klucz użyty do tego szyfrowania.

Szyfr zawiera wiele słów. Jego łączna długość nie przekracza 1024 znaków. Szyfrowaniu podlegały tylko wielkie litery tekstu, zaś odstępy i znaki przestankowe pozostały bez zmiany.

Odszyfruj tekst i umieść jego postać źródłową w pliku z odpowiedziami.

77.3.

1. Podaj liczby wystąpień poszczególnych liter A, B, ..., Z w treści szyfru zawartego w pierwszym wierszu pliku szyfr.txt.
2. Chcąc złamać szyfr Vigenère, nie znając klucza, w pierwszym kroku należy oszacować długość klucza (rozumianą jako liczba znaków). Istnieje przybliżony wzór na szacunkową długość klucza *d* danego szyfru Vigenère’a dla tekstu nad alfabetem 26-literowym. Oszacowanie jest tym lepsze, im dłuższy jest szyfr.

gdzie κo to indeks koincydencji znaków obliczany następująco:

*n* — łączna liczba wystąpień **wszystkich** **liter** w tekście szyfru (nie liczymy odstępów i znaków przestankowych),

, — liczby wystąpień **poszczególnych liter** A, B, …, Z w tekście szyfru.

Wykorzystując powyższe wzory, wyznacz szacunkową długość klucza dla szyfru danego w pierwszym wierszu pliku szyfr.txt i porównaj z dokładną długością klucza umieszczonego w drugim wierszu tego pliku. Wypisz obie wartości, wartość szacunkową zaokrąglij do 2 cyfr po przecinku.

Zadanie 78.

Wiązka zadań *Podpis elektroniczny*

Bajtek otrzymał od przyjaciela 11 jawnych wiadomości. Do każdej wiadomości przyjaciel dołączył podpis elektroniczny.

Podpis elektroniczny jest **zaszyfrowanym skrótem** wiadomości. Przyjaciel Bajtka utworzył skrót za pomocą funkcji *skrot*(), która przekształca dowolną wiadomość w 8-znakowy napis, a następnie zaszyfrował ten skrót algorytmem *A* o sobie tylko znanym kluczu prywatnym (*e*,*n*). Opis obu algorytmów podano poniżej.

Bajtek chciałby być pewien, że zachowano:

* integralność danych (treść nie została zmieniona w trakcie przesyłania),
* uwierzytelnienie nadawcy (nikt się pod nadawcę nie podszył).

W tym celu powinien sprawdzić każdą wiadomość następująco:

* zaprogramować funkcję *skrot*(*wiadomosc*) i za jej pomocą utworzyć skrót wiadomości,
* odszyfrować skrót z podpisu elektronicznego algorytmem *A* przy pomocy ogólnie znanego klucza publicznego (***d***,***n***) o wartościach (**3**,**200**)*,*
* porównać oba skróty: jeśli są identyczne, znaczy to, że wiadomość jest wiarygodna.

Pomóż Bajtkowi sprawdzić, czy otrzymane wiadomości są wiarygodne.

W pliku wiadomosci.txt znajduje się 11 wiadomości, każda w osobnym wierszu. Liczba znaków każdej wiadomości nie przekracza 255. Wiadomości zawierają znaki pojedynczego odstępu, które są integralną częścią informacji.

W pliku podpisy.txt znajduje się 11 wierszy, każdy z nich zawiera 8 liczb całkowitych, stanowiących elementy podpisu elektronicznego jednej wiadomości. Liczby w wierszu oddzielone są pojedynczymi znakami odstępu. Kolejność wierszy podpisów jest zgodna z kolejnością wierszy wiadomości (pierwszy wiersz podpisów odpowiada pierwszej wiadomości, drugi — drugiej itd.)

**Funkcja skrótu *skrot*(*wiadomość*)**

Skrót wiadomości jest 8-znakowym napisem, złożonym z wielkich liter alfabetu angielskiego.

Aby go wyznaczyć, wykonaj następujące kroki:

1. Wpisz do 8-elementowej tablicy *S* kody ASCII znaków słowa ”ALGORYTM”.
2. Treść wiadomości uzupełnij na końcu znakami kropki ’.’ do wielokrotności 8 znaków.
3. Rozpatrz po kolei 8-znakowe porcje treści wiadomości. W zależności od kodów ich znaków aktualizuj wartości elementów w tablicy S. Dla każdej porcji treści wiadomości powtarzaj:

dla *j* =1,2 .. 8 wykonuj

*S*[*j*]  **(***S*[*j*]) + kod znaku na *j*-tej pozycji w bieżącej porcji wiadomości) *mod* 128

1. Zbuduj wynik, wyznaczając jego kolejne znaki na podstawie elementów tablicy S:

*wynik* = ””

dla *j* =1,2 .. 8 wykonuj

*wynik*  *wynik* + char( 65 + *S*[*j*] *mod* 26 )

gdzie: *mod* jest operatorem dzielenia modulo,

funkcja *char*(*kod*) zwraca reprezentację graficzną znaku o podanym kodzie

Otrzymany w ten sposób *wynik* jest skrótem wiadomości.

**Algorytm *A* szyfrowania z kluczem prywatnym (*e*,*n*) i deszyfrowania kluczem publicznym (*d*,*n*)**

Deszyfrowanie polega na wykonaniu operacji *x=(y\*d mod n)*, gdzie za *y* należy przyjąć kolejne liczby tworzące podpis elektroniczny. Tekst wynikowy można otrzymać, składając w jeden napis reprezentacje graficzne kolejnych liczb *x* zgodnie ze standardem ASCII.

Uwaga dla dociekliwych

Zaszyfrowanie algorytmem *A* polegało na wykonywaniu operacji ***y=(x\*e mod n)***, gdzie za *x* należało podstawić kody ASCII kolejnych znaków tekstu źródłowego. Uzyskany w ten sposób ciąg liczb jest podpisem elektronicznym wiadomości. Gdyby ktoś chciał złamać szyfr *A*, czyli wyznaczyć nieznany element *e* klucza prywatnego, musiałby znaleźć taką wartość *e,* względnie pierwszą z *d*, że *e\*d mod n = 1*. Uzasadnienia szukaj w prawach arytmetyki modularnej.

Napisz program rozwiązujący poniższe zadania. Do oceny oddaj plik tekstowyepodpis\_wynik.txt, zawierający odpowiedzi, oraz plik (pliki) zawierający reprezentację komputerową Twojego rozwiązania.

78.1.

Wyznacz skrót **pierwszej** wiadomości z pliku wiadomosci.txt i udokumentuj wyniki kolejnych etapów obliczania tego skrótu. Zapisz w kolejnych wierszach pliku wynikowego:

1. liczbę znaków wiadomości po jej uzupełnieniu do najmniejszej długości o wielokrotności 8 znaków,
2. wartości liczbowe 8 kolejnych bajtów skrótu (elementy tablicy *S*) po przetworzeniu całej wiadomości — wszystkie wartości w jednym wierszu, oddzielone pojedynczymi znakami odstępu,
3. skrót wiadomości w postaci napisu o długości 8, złożonego z wielkich liter alfabetu angielskiego.

78.2.

Odszyfruj skróty wiadomości ze wszystkich podpisów elektronicznych umieszczonych w pliku podpisy.txt, stosując algorytm *A* z kluczem publicznym (*d*,*n*) = (3,200). Zapisz uzyskane skróty w kolejnych, osobnych wierszach pliku z odpowiedziami.

78.3.

Zweryfikuj wiarygodność wszystkich wiadomości i podaj numery wiadomości wiarygodnych. Zapisz w jednym wierszu pliku z odpowiedziami, jako liczby z zakresu 1..11, zgodnie z kolejnością umieszczenia ich w pliku danych, oddzielone pojedynczym znakiem odstępu.

Zadanie 79.

Wiązka zadań *Okręgi*

Okrąg na płaszczyźnie reprezentujemy za pomocą trzech liczb , gdzie oznaczają współrzędne środka okręgu, a — jego promień.

W pliku okregi.txt danych jest 2000 okręgów; każdy zapisany jest w osobnym wierszu, zawierającym trzy liczby rzeczywiste .

Pierwsze **dziewięć wierszy** tego pliku zawiera liczby:

8.000 5.000 40.815

22.623 21.558 50.000

53.551 38.508 25.663

63.429 42.655 18.844

77.352 48.799 20.286

80.274 52.627 16.127

178.318 128.106 23.993

191.501 140.711 8.379

197.073 143.275 13.129

Można zauważyć, że ostatnie siedem okręgów w całości zawiera się w pierwszej ćwiartce układu współrzędnych. Natomiast pierwsze dwa okręgi nie leżą w całości w żadnej ćwiartce.

Napisz program, który umożliwi wykonanie poniższych poleceń. Odpowiedzi do poszczególnych zadań zapisz w pliku tekstowym wyniki\_okregi.txt. Wyniki do każdego zadania poprzedź numerem je oznaczającym.

79.1.

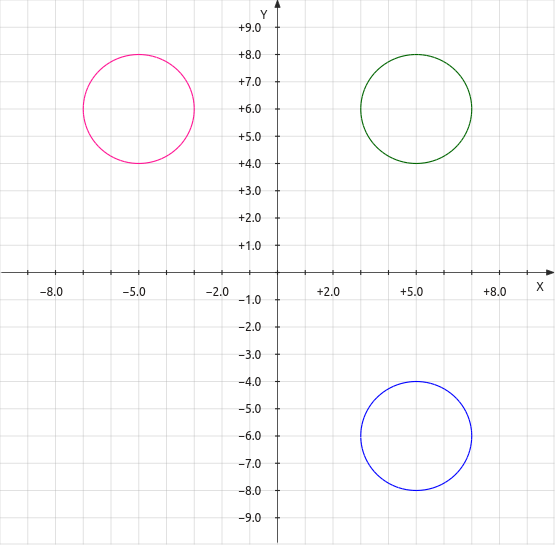
Podaj liczbę okręgów, które całkowicie zawierają się w I, II, III i IV ćwiartce układu współrzędnych. Podaj również liczbę okręgów, które nie zawierają się w całości w żadnej ćwiartce, tzn. mają co najmniej jeden punkt wspólny z jedną z osi Ox lub Oy. Jako odpowiedź wypisz pięć liczb: liczba okręgów I ćwiartki, liczba okręgów II ćwiartki, liczba okręgów III ćwiartki, liczba okręgów IV ćwiartki oraz liczba okręgów, które nie zawierają się w całości w żadnej ćwiartce.

79.2.

Powiemy, że dwa okręgi tworzą *lustrzaną parę*, jeśli jeden z nich powstaje przez odbicie drugiego względem jednej z osi Ox lub Oy. Podaj liczbę lustrzanych par spośród wszystkich okręgów zapisanych w pliku okregi.txt.

Uwagi

1. Układ trzech okręgów powstałych przez odbicia względem osi Ox lub Oy zawiera **dwie** lustrzane pary; np. wśród okręgów o środkach w punktach (-5,6), (5,6), (5,-6)   
   (i o tych samych promieniach) są dokładnie dwie lustrzane pary; zob. poniższy rysunek.

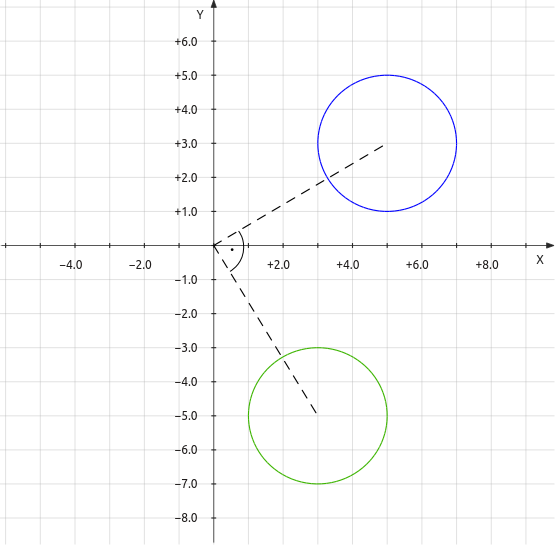


1. Analogicznie do poprzedniego punktu układ czterech okręgów zawiera **cztery** lustrzane pary.

79.3.

Powiemy, że dwa okręgi tworzą *prostopadłą parę*, jeśli jeden z nich powstaje przez obrót drugiego o 90 stopni względem środka układu współrzędnych.

Przykład: okręgi o środkach w punktach (3,-5), (5,3) (i o tych samych promieniach) tworzą parę prostopadłą; zob. rysunek.

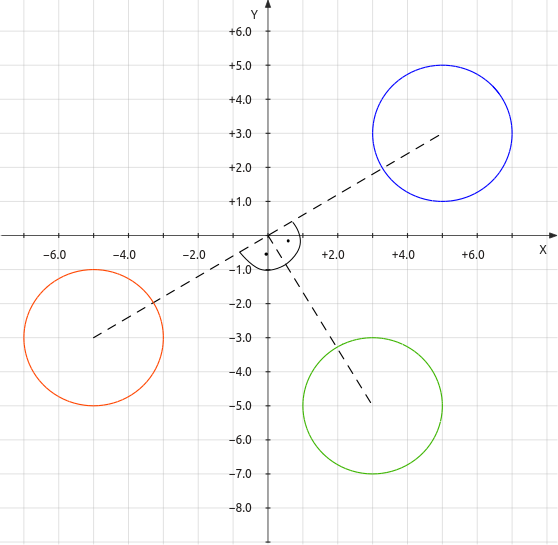


Podaj liczbę prostopadłych par okręgów spośród wszystkich okręgów zapisanych w pliku okregi.txt.

Uwagi

1. układ trzech okręgów powstałych przez obrót o 90 stopni względem środka układu współrzędnych zawiera **dwie** pary prostopadłe;

np. wśród okręgów o środkach (5,3), (3,-5), (-5,-3) (i o tych samych promieniach) są dokładnie dwie pary prostopadłe.



1. Analogicznie do poprzedniego punktu układ czterech okręgów zawiera **cztery** pary prostopadłe.

79.4.

Powiemy, że ciąg okręgów tworzy *łańcuch*, jeśli kolejne okręgi tego ciągu mają ze sobą co najmniej jeden punkt wspólny; przyjmujemy, że ciąg zawierający tylko jeden okrąg również tworzy łańcuch. Można zauważyć, że wśród podanych okręgów mamy dwa łańcuchy: pierwszy o długości 6, a drugi o długości 3.

Znajdź długości wszystkich łańcuchów tworzonych przez okręgi zapisane w wierszach o numerach **od 1 do 1000**. Podaj długość najdłuższego łańcucha.

Zadanie 80.

Wiązka zadań *Trójkąty*

W pliku dane\_trojkaty.txt znajduje się 500 liczb całkowitych dodatnich. Każda liczba jest zapisana w osobnym wierszu, żadna liczba **nie występuje w pliku więcej niż jeden raz**. Liczby podane w pliku dane\_trojkaty.txt to długości odcinków, z których będziemy próbować budować trójkąty.

**Napisz program(-y)**, który pozwoli rozwiązać poniższe zadania. Odpowiedzi zapisz w plikuwyniki\_trojkaty.txt. Odpowiedź do każdego zadania podaj w osobnym wierszu, poprzedzając ją numerem zadania.

80.1.

Wypisz wszystkie trójki **kolejnych** liczb z pliku dane\_trojkaty.txt, które są długościami boków trójkąta prostokątnego.

Przykład

Rozważmy plik składający się z dziesięciu liczb: 8, 7, 4, 3, 5, 9, 12, 13, 85, 84. Wynikiem są dwie trójki liczb: 4, 3, 5 oraz 13, 85, 84. Trójkąt prostokątny tworzą też odcinki o bokach 5, 12 i 13, ale liczby te **nie są kolejnymi** liczbami w podanym pliku.

80.2.

Podaj największy obwód trójkąta, którego boki mają długości równe liczbom występującym w różnych wierszach pliku dane\_trojkaty.txt.

Przykład

Dla pliku składającego się z dziesięciu liczb: 10, 18, 70, 100, 15, 13, 21, 12, 1, 2 wynikiem jest 54, ponieważ trójkąt o największym obwodzie ma boki 18, 15 i 21.

80.3.

Podaj, ile **nieprzystających** trójkątów można utworzyć z odcinków o długościach podanych w pliku dane\_trojkaty.txt.

**Uwaga:** Dwa trójkąty są przystające wtedy i tylko wtedy, gdy trzy boki jednego trójkąta są odpowiednio równe trzem bokom drugiego trójkąta, np. trójkąt o bokach (10, 18, 15) jest przystający z trójkątem o bokach (18, 15, 10).

Przykład

Dla pliku składającego się z dziesięciu liczb: 10, 18, 70, 100, 15, 13, 21, 12, 1, 2 wynikiem jest 21, gdyż z podanych długości odcinków można utworzyć 21 trójkątów o następujących bokach:

(10, 18, 15); (10, 18, 13); (10, 18, 21); (10, 18, 12); (10, 15, 13); (10, 15, 21); (10, 15, 12); (10, 13, 21); (10, 13, 12); (10, 21, 12); (18, 15, 13); (18, 15, 21); (18, 15, 12); (18, 13, 21); (18, 13, 12); (15, 13, 21); (15, 13, 12); (15, 21, 12); (13, 21, 12); (13, 12, 2); (18, 21, 12).

Zadanie 81.

Wiązka zadań *Czworokąty*

Plik wierzcholki.txt zawiera 100 wierszy. W każdym wierszu zapisano 6 liczb całkowitych z przedziału <–100; 100>, będących współrzędnymi trzech różnych punktów: A, B i C w kartezjańskim układzie współrzędnych (odpowiednio xa, ya, xb, yb, xc, yc). Liczby w wierszu są oddzielone pojedynczymi znakami tabulacji.

Podobny plik wierzcholkiTR.txt zawiera również 100 wierszy. W każdym wierszu zapisano 6 liczb całkowitych należących do przedziału <–100; 100>, będących współrzędnymi trzech wierzchołków trójkąta ABC (odpowiednio xa, ya, xb, yb, xc, yc). Liczby w wierszu są oddzielone pojedynczymi znakami tabulacji.

Wykorzystując dane zawarte w plikach oraz dostępne narzędzia informatyczne, rozwiąż poniższe zadania.

Napisz program(-y), za pomocą którego(-ych) uzyskasz odpowiedzi do poniższych zadań. Do oceny oddaj dokument wyniki.txt z zapisanymi odpowiedziami na poszczególne zadania oraz pliki źródłowe programów wykorzystanych do uzyskania rozwiązania.

81.1.

Podaj liczbę wierszy z pliku wierzcholki.txt, w których wszystkie zapisane punkty leżą w I ćwiartce układu współrzędnych i nie należą do osi OX i OY.

81.2.

Podaj liczbę wierszy z pliku wierzcholki.txt, w których zapisane są współrzędne punktów leżących na jednej prostej.

81.3.

Podaj (z pliku wierzcholkiTR.txt) współrzędne wierzchołków trójkąta o największym obwodzie oraz obwód tego trójkąta. Obwód zaokrąglij do dwóch miejsc po przecinku. Uwaga: możesz założyć, że jest tylko jeden taki trójkąt.

81.4.

Dla każdego wiersza z pliku wierzcholkiTR.txt sprawdź, czy punkty zapisane w tym wierszu są wierzchołkami pewnego trójkąta prostokątnego. Podaj liczbę trójkątów prostokątnych zapisanych w tym pliku

**Uwaga:** Takich trójkątów jest więcej niż cztery.

81.5.

Dla każdego wiersza z pliku wierzcholkiTR.txt, w których zapisane są kolejno współrzędne punktów A, B i C, wyznacz współrzędne punktu D, tak aby czworokąt ABCD był równoległobokiem. Podaj współrzędne wszystkich wierzchołków czworokątów ABCD, których punkt D leży na prostej *y*=*x*.