

# Zbiór zadań do przedmiotu Wstęp do Informatyki

Marek Gajęcki

1 października 2025

## Zestaw 1: Proste programy z pętlami

**Zadanie 1.** Trójką prostokątny o bokach wyrażonych liczbami naturalnymi nazywamy Pitagorejskim. Proszę napisać program poszukujący trójkątów Pitagorejskich w których długość przekątnej jest mniejsza od liczby  $N$  wprowadzonej z klawiatury.

**Zadanie 2.** Proszę napisać program odnajdujący cyfry  $a, b, c$  w działaniu:  $abc + abc + abc = bbb$ .

**Zadanie 3.** Proszę napisać program wypisujący elementy ciągu Fibonacciego mniejsze od miliona.

**Zadanie 4.** Proszę znaleźć wyrazy początkowe ciągu zamiast 1,1 o najmniejszej sumie, aby w ciągu analogicznym do ciągu Fibonacciego wystąpił wyraz równy numerowi bieżącego roku.

**Zadanie 5.** Proszę napisać program sprawdzający czy istnieje spójny podciąg ciągu Fibonacciego o zadanej sumie.

**Zadanie 6.** Pierwiastek całkowitoliczbowy z liczby naturalnej to część całkowita z pierwiastka z tej liczby. Proszę napisać program obliczający taki pierwiastek korzystając z zależności  $1 + 3 + 5 + \dots = n^2$ .

**Zadanie 7.** Proszę napisać program wyznaczający pierwiastek kwadratowy ze wzoru Newtona.

**Zadanie 8.** Proszę zmodyfikować wzór Newtona aby program z poprzedniego zadania obliczał pierwiastek stopnia 3.

**Zadanie 9.** Proszę napisać program rozwiązujący równanie  $x^x = 2024$  metodą bisekcji.

**Zadanie 10.** Proszę napisać program wczytujący liczbę naturalną i odpowiadający na pytanie, czy liczba ta jest iloczynem dowolnych dwóch kolejnych wyrazów ciągu Fibonacciego.

**Zadanie 11.** Proszę napisać program sprawdzający czy zadana liczba jest pierwszą.

**Zadanie 12.** Proszę napisać program wypisujący dzielniki liczby.

**Zadanie 13.** Proszę napisać program wypisujący rozkład liczby na czynniki pierwsze.

**Zadanie 14.** Liczba doskonała to liczba równa sumie swoich dzielników właściwych (mniejszych od niej samej), na przykład  $6 = 1 + 2 + 3$ . Proszę napisać program wyszukujący liczby doskonałe mniejsze od miliona.

**Zadanie 15.** Dwie różne liczby nazywamy zaprzyjaźnionymi gdy każda jest równa sumie dzielników właściwych drugiej liczby, na przykład 220 i 284. Proszę napisać program wyszukujący liczby zaprzyjaźnione mniejsze od miliona.

**Zadanie 16.** Proszę napisać program wyznaczający największy wspólny dzielnik 3 zadanych liczb naturalnych.

**Zadanie 17.** Proszę napisać program wyznaczający najmniejszą wspólną wielokrotność 3 zadanych liczb naturalnych.

**Zadanie 18.** Proszę napisać program obliczający wartości  $\cos(x)$  z rozwinięcia w szereg Maclaurina.  
$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

**Zadanie 19.** Nieskończony iloczyn  $\sqrt{0.5} * \sqrt{0.5 + 0.5 * \sqrt{0.5}} * \sqrt{0.5 + 0.5 * \sqrt{0.5 + 0.5 * \sqrt{0.5}}} * \dots$  ma wartość  $2/\pi$ . Proszę napisać program korzystający z tej zależności i wyznaczający wartość  $\pi$ .

**Zadanie 20.** Dany jest ciąg określony wzorem:  $A_{n+1} = (A_n \bmod 2) * (3 * A_n + 1) + (1 - A_n \bmod 2) * A_n / 2$ . Startując z dowolnej liczby naturalnej  $> 1$  ciąg ten osiąga wartość 1. Proszę napisać program, który znajdzie wyraz początkowy z przedziału 2-10000 dla którego wartość 1 jest osiągalna po największej liczbie kroków.

**Zadanie 21.** Dla ciągu z poprzedniego zadania proszę znaleźć najmniejszy wyraz początkowy  $N$ , dla którego ciąg osiąga wartość 1 dokładnie po  $N$  krokach.

**Zadanie 22.** Proszę napisać program wyznaczający wartość do której zmierza iloraz dwóch kolejnych wyrazów ciągu Fibonacciego. Wyznaczyć te ilorazy dla różnych wartości dwóch początkowych wyrazów ciągu.

**Zadanie 23.** Proszę napisać program wyznaczający wartość liczby  $e$  korzystając z zależności:  $e = 1/0! + 1/1! + 1/2! + 1/3! + \dots$

**Zadanie 24.** Dane są ciągi:  $A_{n+1} = \sqrt{A_n * B_n}$  oraz  $B_{n+1} = (A_n + B_n)/2.0$ . Ciągi te są zbieżne do wspólnej granicy nazywanej średnią arytmetyczno-geometryczną. Proszę napisać program wyznaczający średnią arytmetyczno-geometryczną dwóch liczb naturalnych.

**Zadanie 25.** Pewne liczby pierwsze są palindromami i pozostają liczbami pierwszymi pomimo pozbawiania ich skrajnych cyfr. Na przykład:  $71317 \rightarrow 131 \rightarrow 3$ . Proszę napisać program, który wypisuje wszystkie takie liczby mniejsze od  $N$ .

**Zadanie 26.** Proszę napisać program wczytujący liczbę naturalną i odpowiadający na pytanie, czy liczba ta jest iloczynem dowolnych dwóch wyrazów ciągu Fibonacciego.

**Zadanie 27.** Proszę napisać program wczytujący trzy liczby naturalne  $a, b, n$  i wypisujący rozwinięcie dziesiętne ułamka  $a/b$  z dokładnością do  $n$  miejsc po kropce dziesiętnej. ( $n$  jest rzędu 100)

**Zadanie 28.** Proszę napisać program wczytujący liczbę naturalną i odpowiadający na pytanie, czy liczba naturalna jest palindromem, a następnie czy jest palindromem w systemie dwójkowym.

**Zadanie 29.** Proszę napisać program wczytujący liczbę naturalną i rozkładający ją na iloczyn 2 liczb o najmniejszej różnicy. Na przykład:  $30 = 5 * 6$ ,  $120 = 10 * 12$ .

**Zadanie 30.** Proszę napisać program wczytujący liczbę naturalną i odpowiadający na pytanie, czy liczba ta jest wielokrotnością dowolnego wyrazu ciągu danego wzorem  $A_n = n * n + n + 1$ .

**Zadanie 31.** Proszę napisać program, który oblicza pole figury pod wykresem funkcji  $y = 1/x$  w przedziale od 1 do  $k$ , metodą prostokątów.

**Zadanie 32.** Proszę napisać program wczytujący liczbę naturalną i odpowiadający na pytanie, czy liczba ta jest wielokrotnością dowolnego wyrazu ciągu danego wzorem  $A_n = 3 * A_{n-1} + 1$ , a pierwszy wyraz jest równy 2.

**Zadanie 33.** Proszę napisać program wczytujący liczbę naturalną i odpowiadający na pytanie, czy jej cyfry stanowią ciąg rosnący.

**Zadanie 34.** Proszę napisać program wczytujący liczbę naturalną i odpowiadający na pytanie, czy jej cyfry stanowią ciąg geometryczny.

**Zadanie 35.** Proszę napisać program wczytujący liczbę naturalną i odpowiadający na pytanie, czy liczba ta zawiera cyfrę równą liczbie swoich cyfr.

**Zadanie 36.** Proszę napisać program wczytujący liczbę naturalną i odpowiadający na pytanie, czy liczba ta zakończona jest unikalną cyfrą.

**Zadanie 37.** Proszę napisać program wyliczający pierwiastek równania  $x^x = 2020$  metodą stycznych.

**Zadanie 38.** Mamy dane dwa ciągi  $A, B$  o następujących zależnościach:

A:  $a_0 = 0, a_1 = 1, a_n = a_{n-1} - b_{n-1} * a_{n-2}$

B:  $b_0 = 2, b_n = b_{n-1} + 2 * a_{n-1}$

Proszę napisać program, który czyta liczby typu int ze standardowego wejścia i tak długo jak liczby te są kolejnymi wyrazami ciągu  $A_n$  (tj.  $a_0, a_1, a_2, \dots$ ) wypisuje na standardowe wyjście wyrazy drugiego ciągu  $B_n$  (tj.  $b_0, b_1, b_2, \dots$ ).

**Zadanie 39.** Proszę napisać program, który wczytuje wprowadzany z klawiatury ciąg liczb naturalnych zakończonych zerem stanowiącym wyłącznie znacznik końca danych. Program powinien wypisać te elementy ciągu które są równe średniej arytmetycznej z 4 najbliższych sąsiadów. Na przykład dla ciągu: 2,3,2,7,1,2,4,8,5,2,2,5,7,9,3,1,0 powinny zostać wypisane liczby: 4,5. Można założyć, że w ciągu znajduje się co najmniej 5 elementów.

**Zadanie 40.** Napisać program, który wyznacza liczbę zer jakimi kończy się liczba  $N!$  Program powinien działać dla  $N$  rzędu  $10^{100}$ .

**Zadanie 41.** Proszę napisać funkcję, która sprawdza czy napis składający się z: liter a..z, operatorów dodawania i mnożenia oraz nawiasów jest poprawnym wyrażeniem arytmetycznym.

**Zadanie 42.** Napisać program, który wyznacza ostatnią niezerową cyfrę liczby  $N!$ . Program powinien działać dla  $N$  rzędu  $10^{100}$ . Komentarz: Rozwiązanie tego zadania w języku Python jest proste, trochę większym wyzwaniem jest rozwiązanie w języku C/C++

**Zadanie 43.** Proszę napisać funkcję, która zwraca wartość True gdy dwie liczby są zbudowane z tych samych cyfr (na przykład: 123 i 231, 5749 i 4597) i wartość False w przeciwnym przypadku.

**Zadanie 44.** Dla pewnej  $N$ -cyfrowej liczby naturalnej obliczono sumę  $N$ -tych potęg cyfr tej liczby otrzymując kolejną liczbę  $N$ -cyfrową. Na przykład dla liczb: 354, 543, 600, ... suma ta wynosi 216. Niestety pierwotna liczba zaginęła ale wiadomo, że była to największa z możliwych takich liczb. Proszę napisać program, który na podstawie zachowanej sumy wyznaczy pierwotną liczbę

**Zadanie 45.** Liczbę pierwszą będącą palindromem nazywamy “palindromem pierwszym”. Liczbę nazywamy “super palindromem pierwszym” jeżeli podczas odrzucania parami skrajnych cyfr do końca pozostaje ona palindromem pierwszym. Na przykład, liczba 373929373 jest super palindromem pierwszym bo 373929373, 7392937, 39293, 929, 2 są palindromami pierwszymi. Początkowymi super palindromami pierwszymi są: 2, 3, 5, 7, 11, 131, 151. Proszę napisać program, który wylicza ile jest super palindromów pierwszych mniejszych od zadanej liczby  $n$ .

**Zadanie 46.** Dane są dwie liczby naturalne,  $m$  i  $n$ . Proszę napisać program, który wyznacza sumę  $n$  kolejnych cyfr po przecinku rozwinięcia dziesiętnego liczby  $\sqrt[n]{m}$

**Zadanie 47.** Mając daną dodatnią liczbę całkowitą  $N$ , stwórzmy nową liczbę dodając kwadraty cyfr liczby  $N$ . Można udowodnić, że postępując w ten sposób wielokrotnie otrzymamy w końcu wynik 1 lub 4. Przykład:  $13 = 1^2 + 3^2 = 1 + 9 = 10$  (Krok 1)  $10 = 1^2 + 0^2 = 1 + 0 = 1$  (Krok 2, kończymy iterację ponieważ uzyskaliśmy liczbę 1) Jeżeli w opisanej powyżej procedurze uzyskamy wynik 1, to liczbę  $N$  nazywamy “jednokwadratową”. Proszę napisać program, który znajduje  $K$ -tą liczbę w zadanym przedziale  $[L, U]$ , która jest jednocześnie jednokwadratowa i pierwsza.

**Zadanie 48.** Wybieramy dodatnią liczbę całkowitą  $X$ . Z liczby  $X$  wykreślamy ostatnią cyfrę. Postępujemy tak, aż usuniemy wszystkie cyfry liczby  $X$ . Następnie sumujemy wszystkie powstałe w ten sposób liczby, włączając liczbę  $X$ . Na przykład, jeżeli wybraliśmy  $X = 1234$  to w kolejnych krokach otrzymamy odpowiednio liczby 1234, 123, 12, 1. Ich suma to 1370. Mamy daną liczbę całkowitą dodatnią  $S$ . Proszę napisać program, który znajduje liczbę  $X$  taką, że powyżej opisana procedura daje sumę  $S$ . Można pokazać, że dla dowolnej dodatniej liczby  $S$  istnieje co najwyżej jedna taka wartość  $X$ . Jeżeli nie ma takiego  $X$  program powinien wypisać -1.

**Zadanie 49.** Proszę napisać program wyznaczający najmniejszą liczbę pierwszą o sumie cyfr równej  $N$ , której cyfry są w porządku rosnącym

**Zadanie 50.** Proszę znaleźć najmniejszą liczbę pierwszą, której suma cyfr wynosi 101, a cyfry są w porządku nierosnącym

**Zadanie 51.** Pewnych liczb nie można przedstawić jako sumy elementów spójnych fragmentów ciągu Fibonacciego, np. 9, 14, 15, 17, 22. Proszę napisać program, który wczytuje liczbę naturalną  $n$ , wylicza i wypisuje następną taką liczbę większą od  $n$ . Można założyć, że  $0 < n < 1000$ .

**Zadanie 52.** Dane są dwie liczby naturalne z których budujemy trzecią liczbę. W budowanej liczbie muszą wystąpić wszystkie cyfry występujące w liczbach wejściowych. Wzajemna kolejność cyfr każdej z liczb wejściowych musi być zachowana. Na przykład mając liczby 123 i 75 możemy zbudować liczby 12375, 17523, 75123, 17253, itd. Proszę napisać funkcję która wyznaczy ile liczb pierwszych można zbudować z dwóch zadanych liczb.

**Zadanie 53.** Liczba Smitha to taka, której suma cyfr jest równa sumie cyfr wszystkich liczb występujących w jej rozkładzie na czynniki pierwsze. Na przykład:  $85 = 5 * 17$ ,  $8 + 5 = 5 + 1 + 7$ . Proszę napisać program wypisujący liczby Smitha mniejsze od  $10^6$ .

**Zadanie 54.** Liczba dwu-trzy-piętkowa w rozkładzie na czynniki pierwsze nie posiada innych czynników niż 2, 3, 5. Jedyneką też jest taką liczbą. Proszę napisać program, który wylicza ile takich liczb znajduje się w przedziale od 1 do  $N$  włącznie.

**Zadanie 55.** Proszę napisać program wczytujący dwie liczby naturalne  $a, b$  i wypisujący rozwinięcie dziesiętne ułamka  $a/b$  uwzględniając ułamki okresowe. Na przykład  $2/3 = 0.(6)$ ,  $1/5 = 0.2$ ,  $1/6 = 0.1(6)$ ,  $1/7 = 0.(142857)$

**Zadanie 56.** Dwie liczby naturalne są różno-cyfrowe jeżeli nie posiadają żadnej wspólnej cyfry. Proszę napisać program, który wczytuje dwie liczby naturalne i poszukuje najmniejszej podstawy systemu (w zakresie  $2 - 16$ ) w którym liczby są różno-cyfrowe. Program powinien wypisać znalezioną podstawę, jeżeli podstawa taka nie istnieje należy wypisać komunikat o jej braku. Na przykład: dla liczb 123 i 522 odpowiedzią jest podstawa 11 bo  $123_{(10)} = 102_{(11)}$  i  $522_{(10)} = 435_{(11)}$ .

**Zadanie 57.** „Obcięcie” liczby naturalnej polega na usunięciu z niej  $M$  początkowych i  $N$  końcowych cyfr, gdzie  $M, N \geq 0$ . Proszę napisać funkcję, która dla danej liczby naturalnej  $K$  zwraca największą liczbę pierwszą o różnych cyfrach jaką można uzyskać z obcięcia liczby  $K$  albo 0 gdy taka liczba nie istnieje. Na przykład dla liczby 1202742516 spośród obciętych liczb pierwszych: 2, 5, 7, 251, 2027 liczbą spełniającą warunek jest liczba 251.

**Zadanie 58.** Liczbę nazywamy iloczynowo-pierwszą jeżeli iloczyn jej cyfr w systemie o podstawie  $b$  jest liczbą pierwszą. Na przykład: 13 jest liczbą iloczynowo-pierwszą w systemie dziesiętnym, bo  $1 * 3 = 3$  16 jest liczbą iloczynowo-pierwszą w systemie trójkowym, bo  $16 = 121_{(3)}$ ,  $1 * 2 * 1 = 2$ . W liczbie naturalnej możemy dokonywać rotacji jej cyfr, np. 1428, 4281, 2814, 8142 albo 209, 092, 920. Proszę napisać funkcję, która dla danej liczby naturalnej  $N$ , zwróci najmniejszą podstawę systemu (z zakresu 2-16) w którym przynajmniej jedna z rotowanych liczb jest iloczynowo-pierwsza albo wartość None gdy taka podstawa nie istnieje.

**Zadanie 59.** Proszę napisać program znajdujący jak najwięcej liczb  $N$ -cyfrowych dla których suma  $N$ -tych potęg cyfr liczby jest równa tej liczbie, np.  $153 = 1^3 + 5^3 + 3^3$ .

**Zadanie 60.** Tylko 7 liczb pierwszych spełnia warunek z poprzedniego zadania. Proszę napisać program znajdujący wszystkie te liczby.

**Zadanie 61.** Dana jest liczba naturalna  $N$ , którą zapisujemy w systemie o podstawie od 2 do 16. Tak zapisaną liczbę ”rozcinamy” w dowolnym miejscu na dwa kawałki, a powstałe w ten sposób liczby mnożymy. Proszę napisać funkcję, która dla liczby  $N$  zwróci najmniejszą podstawę systemu, w którym można uzyskać największy iloczyn, a dodatkowo obie liczby powstałe z podziału są względnie pierwsze. Na przykład: liczba  $N = 202$  w systemie o podstawie 6 ma postać  $534_{(6)}$ . Możliwe podziały w tym systemie o względnie pierwszych czynnikach to:  $5_{(6)} * 34_{(6)}$  i  $53_{(6)} * 4_{(6)}$  co w systemie o podstawie 10 odpowiada iloczynom:  $5 * 22 = 110$  i  $33 * 4 = 132$ . Ten drugi jest największym możliwym do osiągnięcia, a zatem funkcja powinna zwrócić wartość podstawy równą 6.

**Zadanie 62.** Dwie liczby naturalne  $A, B$  można ”skleić” ze sobą jeżeli 2 lub 3 ostatnie cyfry liczby  $A$  pokrywają się z odpowiednio z pierwszymi 2 lub 3 cyframi liczby  $B$ . Na przykład: 2571 i 710, 12345678 i 678999. Siła z jaką skleja się liczby to liczba złożona z dopasowanych cyfr, dla powyższych przykładów to odpowiednio: 71 i 678. Dany jest cykl złożony z  $N$  **8 cyfrowych** liczb, zawarty w tablicy  $T$ . Proszę napisać funkcję, która w takim cyklu znajdzie fragment o największej, sumarycznej sile sklejenia. Funkcja powinna zwrócić siłę sklejenia znalezionego cyklu. W przypadku gdy żadnych dwóch kolejnych liczb nie uda się skleić, funkcja powinna zwrócić wartość 0. Jeżeli uda się skleić cały cykl funkcja powinna zwrócić  $-1$ .

Dla tablicy:  $T = [79000023, 23111134, 55555555, 66666104, 10467700, 8888879]$

Funkcja powinna zwrócić wartość 104 (elementy o indeksach 3,4), inny ciąg złożony z elementów o indeksach 5,0,1 ma siłę sklejenia  $79+23=102$

**Zadanie 63.** Dana jest liczba naturalna  $N$ . Proszę napisać funkcję, która zwróci maksymalną długość powtarzającej się sekwencji cyfr w tej liczbie. Jeżeli w liczbie nic się nie powtarza zwracamy 0. Przykłady powtarzających się sekwencji: 2024 powtarza się 2, zwracamy 1, 123456 nic się nie powtarza, zwracamy 0, 1234568234 powtarza się 234, zwracamy 3, 222222200 najdłuższa powtarzająca się sekwencja to 2222, zwracamy 4.

## Zestaw 2: Tablice jednowymiarowe

**Zadanie 64.** Napisać funkcję zamieniającą i wypisującą liczbę naturalną na system o podstawie 2-16.

**Zadanie 65.** Napisać funkcję sprawdzającą czy dwie liczby naturalne są one zbudowane z takich samych cyfr, np. 123 i 321, 1255 i 5125, 11000 i 10001.

**Zadanie 66.** Napisać program generujący i wypisujący liczby pierwsze mniejsze od  $N$  metodą Sita Eratostenesa.

**Zadanie 67.** Proszę napisać program obliczający i wypisujący stałą  $e$  z rozwinięcia w szereg  $e = 1/0! + 1/1! + 1/2! + 1/3! + \dots$  z dokładnością  $N$  cyfr dziesiętnych ( $N$  jest rzędu 1000).

**Zadanie 68.** Napisać program, który wczytuje wprowadzany z klawiatury ciąg liczb naturalnych zakończonych zerem stanowiącym wyłącznie znacznik końca danych. Program powinien wypisać 10 co do wielkości wartość, jaka wystąpiła w ciągu. Można założyć, że w ciągu znajduje się wystarczająca liczba elementów.

**Zadanie 69.** Napisać program wypełniający  $N$ -elementową tablicę  $T$  liczbami naturalnymi 1-1000 i sprawdzający czy każdy element tablicy zawiera co najmniej jedną cyfrę nieparzystą.

**Zadanie 70.** Napisać program wypełniający  $N$ -elementową tablicę  $T$  liczbami pseudolosowymi z zakresu 1-1000 i sprawdzający czy istnieje element tablicy zawierający wyłącznie cyfry nieparzyste.

**Zadanie 71.** Dana jest  $N$ -elementowa tablica  $T$  zawierająca liczby naturalne. W tablicy możemy przeskoczyć z pola o indeksie  $k$  o  $n$  pól w prawo jeżeli wartość  $n$  jest czynnikiem pierwszym liczby  $T[k]$ . Napisać funkcję sprawdzającą czy jest możliwe przejście z pierwszego pola tablicy na ostatnie pole.

**Zadanie 72.** Napisać funkcję, która dla  $N$ -elementowej tablicy  $T$  wypełnionej liczbami naturalnym wyznacza długość najdłuższego, spójnego podciągu rosnącego.

**Zadanie 73.** Napisać funkcję, która dla  $N$ -elementowej tablicy  $T$  wypełnionej liczbami naturalnym wyznacza długość najdłuższego, spójnego podciągu arytmetycznego.

**Zadanie 74.** Napisać funkcję, która dla  $N$ -elementowej tablicy  $T$  wypełnionej liczbami naturalnym wyznacza długość najdłuższego, spójnego podciągu geometrycznego.

**Zadanie 75.** Proszę napisać program, który wypełnia  $N$ -elementową tablicę  $T$  pseudolosowymi liczbami nieparzystymi z zakresu  $[1..99]$ , a następnie wyznacza i wypisuje różnicę pomiędzy długością najdłuższego znajdującego się w niej ciągu arytmetycznego o dodatniej różnicy, a długością najdłuższego ciągu arytmetycznego o ujemnej różnicy, przy założeniu, że kolejnymi wyrazami ciągu są elementy tablicy o kolejnych indeksach.

**Zadanie 76.** Proszę napisać program, który wypełnia  $N$ -elementową tablicę  $T$  trzycyfrowymi liczbami pseudolosowymi, a następnie wyznacza i wypisuje długość najdłuższego podciągu spójnego znajdującego się w tablicy dla którego w tablicy występuje również rewers tego ciągu. Na przykład dla tablicy:  $t = [2, 9, 3, 1, 7, 11, 9, 6, 7, 7, 1, 3, 9, 12, 15]$  odpowiedzią jest liczba 4.

**Zadanie 77.** Napisać program wyznaczający na drodze eksperymentu prawdopodobieństwo tego, że w grupie  $N$  przypadkowo spotkanych osób, co najmniej dwie urodziły się tego samego dnia roku. Wyznaczyć wartości prawdopodobieństwa dla  $N$  z zakresu 20-40.

**Zadanie 78.** Dana jest duża tablica  $T$ . Proszę napisać funkcję, która zwraca informację czy w tablicy zachodzi następujący warunek: „wszystkie elementy, których indeks jest elementem ciągu Fibonacciego są liczbami złożonymi, a wśród pozostałych przynajmniej jedna jest liczbą pierwszą”

**Zadanie 79.** Mamy zdefiniowaną  $n$ -elementową tablicę liczb całkowitych. Proszę napisać funkcję zwracającą wartość typu bool oznaczającą, czy w tablicy istnieje dokładnie jeden element najmniejszy i dokładnie jeden element największy (liczba elementów najmniejszych oznacza liczbę takich elementów o tej samej wartości).

**Zadanie 80.** Dane są dwie  $N$ -elementowe tablice  $t1$  i  $t2$  zawierające liczby naturalne. Z wartości w obu tablicach możemy tworzyć sumy. „Poprawna” suma to taka, która zawiera co najmniej jeden element (z tablicy  $t1$  lub  $t2$ ) o każdym indeksie. Na przykład dla tablic:  $t1 = [1,3,2,4]$  i  $t2 = [9,7,4,8]$  poprawnymi sumami są na przykład  $1+3+2+4$ ,  $9+7+4+8$ ,  $1+7+3+8$ ,  $1+9+7+2+4+8$ . Proszę napisać funkcję generującą i wypisującą wszystkie poprawne sumy, które są liczbami pierwszymi. Do funkcji należy przekazać dwie tablice, funkcja powinna zwrócić liczbę znalezionych i wypisanych sum.

**Zadanie 81.** Dana jest  $N$ -elementowa tablica  $t$  wypełniona liczbami naturalnymi. Proszę napisać funkcję, która zwraca długość najdłuższego spójnego podciągu będącego palindromem złożonym wyłącznie z liczb nieparzystych. Do funkcji należy przekazać tablicę, funkcja powinna zwrócić długość znalezionego podciągu lub wartość 0 jeżeli taki podciąg nie istnieje.

**Zadanie 82.** Dana jest  $N$ -elementowa tablica  $t$  wypełniona liczbami naturalnymi. Proszę napisać funkcję, która zwraca długość najdłuższego, spójnego podciągu rosnącego dla którego suma jego elementów jest równa sumie indeksów tych elementów. Do funkcji należy przekazać tablicę, funkcja powinna zwrócić długość znalezionego podciągu lub wartość 0 jeżeli taki podciąg nie istnieje.

**Zadanie 83.** Dana jest  $N$ -elementowa tablica  $t$  zawierająca liczby naturalne mniejsze od 1000. Proszę napisać funkcję, która zwraca długość najdłuższego, spójnego fragmentu tablicy, dla którego w iloczynie jego elementów każdy czynnik pierwszy występuje co najwyżej raz. Na przykład dla tablicy  $t=[2,23,33,35,7,4,6,7,5,11,13,22]$  wynikiem jest wartość 5.

**Zadanie 84.** Proszę napisać program obliczający i wypisujący wartość  $N!$  dla  $N$  z zakresu od 1 do 1000. To zadanie należy napisać w języku C/C++.

**Zadanie 85.** Napisać program, który wyznacza  $n$ -tą cyfrę po przecinku rozwinięcia dziesiętnego wartości  $\sqrt{2}$ . Program powinien działać poprawnie dla  $n < 100$ .

**Zadanie 86.** Dana jest tablica  $T$  zawierająca liczby wymierne reprezentowane w postaci ułamków. Ułamki reprezentowane są w postaci krotek składających się z licznika i mianownika. Proszę napisać funkcję zwracającą długość najdłuższego spójnego podciągu, którego elementy stanowią ciąg geometryczny. W przypadku gdy w tablicy nie ma takiego podciągu dłuższego niż 2 elementy, funkcja powinna zwrócić wartość 0. Można założyć, że tablica wejściowa liczy więcej niż 2 elementy.

**Zadanie 87.** Napis nazywamy wielokrotnym, jeżeli powstał przez  $n$ -krotne ( $n > 1$ ) powtórzenie innego napisu o długości co najmniej 1. Przykłady napisów wielokrotnych:  $ABCABCABC$ ,  $AAAA$ ,  $ABAABA$ . Dana jest tablica  $T[N]$  zawierająca napisy. Proszę napisać funkcję  $\text{multi}(T)$ , która zwraca długość najdłuższego napisu wielokrotnego występującego w tablicy  $T$  lub wartość 0, jeżeli takiego napisu nie ma w tablicy.

**Zadanie 88.** Dana jest tablica  $T[N]$  wypełniona niepowtarzającymi się liczbami naturalnymi. Proszę napisać funkcję  $\text{trojki}(T)$  która zlicza wszystkie trójki liczb, które spełniają następujące warunki:  
(1) największym wspólnym dzielnikiem trzech liczb jest liczba 1,

(2) pomiędzy dwoma kolejnymi elementami trójki może być co najwyżej jedna przerwa. Funkcja powinna zwrócić liczbę znalezionych trójek.

Przykładowe wywołania funkcji:

```
print(trojki([2,4,6,7,8,10,12])) # 0 trójek
print(trojki([2,3,4,6,7,8,10])) # 1 trójka (3,4,7)
print(trojki([2,4,3,6,5])) # 2 trójki (2,3,5), (4,3,5)
print(trojki([2,3,4,5,6,8,7])) # 5 trójek (2,3,5), (3,4,5), (3,5,8), (5,6,7), (5,8,7)
```

**Zadanie 89.** Dwie liczby naturalne są 4-zgodne, jeżeli po zapisaniu ich w systemie o podstawie 4, zbiory cyfr występujące w liczbie są identyczne. Na przykład:

$$13 = 31_{(4)} \text{ i } 23 = 113_{(4)}$$

$$18 = 102_{(4)} \text{ i } 33 = 201_{(4)}$$

$$107 = 1223_{(4)} \text{ i } 57 = 321_{(4)}.$$

Dana jest tablica  $T[N]$  zawierająca  $N$  liczb naturalnych. Proszę napisać funkcję, która zwraca długość najdłuższego podciągu (niekoniecznie spójnego) złożonego z liczb 4-zgodnych.

**Zadanie 90.** Dana jest  $N$ -elementowa tablica  $T$ , zawierająca liczby. Proszę napisać funkcję, która zwróci indeks największej liczby, która jest iloczynem wszystkich liczb pierwszych leżących w tablicy na indeksach mniejszych od niej, lub `None` jeżeli taka liczba nie istnieje.

**Zadanie 91.** Proszę napisać funkcję, która oblicza ile dni dzieli dwie daty. Na przykład daty 19.05.1964 i 21.06.1970 dzieli 2224 dni. Do funkcji należy przekazać obie daty, funkcja powinna zwrócić liczbę dni pomiędzy tymi datami. Daty mogą pochodzić z lat 1900-2100.

**Zadanie 92.** Korzystając z funkcji z poprzedniego zadania oraz wiedząc, że 1 stycznia 1900 roku był poniedziałek, proszę napisać program obliczający jaki dzień tygodnia przypada na określoną datę

**Zadanie 93.** Dana jest liczba naturalna o niepowtarzających się cyfrach pośród których nie ma zera. Ile różnych liczb podzielnych np. przez 7 można otrzymać poprzez wykreślenie dowolnych cyfr w tej liczbie. Np. dla 2315 będą to 21, 35, 231, 315.

**Zadanie 94.** Proszę napisać program, który wyznacza wartość pierwiastka kwadratowego z liczby naturalnej  $x$  z dokładnością do  $N$  miejsc dziesiętnych po przecinku. Program powinien działać poprawnie dla  $x < 10^8$  i  $N < 100$ .

**Zadanie 95.** Dwie liczby naturalne większe od 1 są zgodne jeżeli dzielą się przez te same liczby pierwsze. Przykładem zgodnych liczb są pary: (6, 24), (40, 50), (13, 169), (44, 242),

nie są zgodne np. pary: (6, 8), (40, 60), (13, 39), (44, 99)

Tablica  $T$  została wypełniona liczbami z zakresu  $[2..999]$ . Sąsiadami pola o indeksie  $i$  w tablicy są pola o indeksie  $j$ , gdy  $abs(i - j) < 3$ . Proszę napisać funkcję *zgodne*( $T$ ), która dla tak wypełnionej tablicy zwraca liczbę elementów mających przynajmniej jednego zgodnego sąsiada. Po wykonaniu funkcji tablica nie musi pozostać nie zmieniona. Na przykład dla tablicy:  $T = [2, 3, 4, 5, 7, 6, 23, 24, 12, 13, 14, 15, 16, 45]$  funkcja powinna zwrócić wartość 7 (są to liczby 2, 4, 6, 24, 12, 15, 45).

**Zadanie 96.** Dana jest tablica  $T$  zawierająca liczby naturalne. W tablicy na kolejnych pozycjach ukryto pewien ciąg liczb o długości co najmniej 3 elementów. Aby ułatwić odnalezienie tego ciągu, zaraz za nim umieszczono ten sam ciąg, ale każdy z jego elementów pomnożono przez pewną liczbę. Proszę napisać funkcję *sequence*( $T$ ) która odnajdzie ukryty ciąg. Funkcja powinna zwrócić indeksy pierwszego i ostatniego elementu ukrytego ciągu. Na przykład dla ciągu: 2,5,7,3,2,3,5,7,6,9,15,21,17,19,23,2,6,4,8,3,5,7,1,3,2 funkcja zwróci 4,7

**Zadanie 97.** Dana jest tablica  $T$  zawierająca liczby wymierne reprezentowane w postaci ułamków. Ułamki reprezentowane są w postaci krotek składających się z licznika i mianownika. Proszę napisać funkcję `longest(T)`, zwracającą długość najdłuższego spójnego podciągu, którego elementy stanowią ciąg geometryczny. W przypadku, gdy w tablicy nie ma ciągu dłuższego niż 2 elementy, funkcja powinna zwrócić wartość 0.

Przykłady:

```
print(longest( [(0,2),(1,2),(2,2),(4,2),(4,1),(5,1)] ) ) # wypisze 4
print(longest( [(1,2),(-1,2),(1,2),(1,2),(1,3),(1,2)] ) ) # wypisze 3
print(longest( [(3,18),(-1,6),(7,42),(-1,6),(5,30),(-1,6)] ) ) # wypisze 6
print(longest( [(1,2),(2,3),(3,4),(4,5),(5,6)] ) ) # wypisze 0
```

## Zestaw 3: Tablice o większej liczbie wymiarów

**Zadanie 98.** Dana jest tablica  $T[N][N]$ . Proszę napisać funkcję wypełniającą tablicę kolejnymi liczbami naturalnymi po spirali.

**Zadanie 99.** Dana jest tablica  $T[N][N]$  wypełniona liczbami naturalnymi. Proszę napisać funkcję, która odpowiada na pytanie, czy w każdym wierszu tablicy występuje co najmniej jedna liczba złożona wyłącznie z nieparzystych cyfr.

**Zadanie 100.** Dana jest tablica  $T[N][N]$  wypełniona liczbami naturalnymi. Proszę napisać funkcję, która odpowiada na pytanie, czy istnieje wiersz w tablicy w którym każda z liczb zawiera przynajmniej jedną cyfrę parzystą.

**Zadanie 101.** Dana jest tablica  $T[N][N]$  wypełniona liczbami naturalnymi. Proszę napisać funkcję, która zwraca wiersz i kolumnę dowolnego elementu, dla którego iloraz sumy elementów w kolumnie w którym leży element do sumy elementów wiersza w którym leży element jest największa.

**Zadanie 102.** Poprzednie zadanie z tablicą wypełnioną liczbami całkowitymi.

**Zadanie 103.** Dane są dwie tablice mogące pomieścić taką samą liczbę elementów:  $T1[N][N]$  i  $T2[M]$ , gdzie  $M=N*N$ . W każdym wierszu tablicy  $T1$  znajdują się uporządkowane rosnąco (w obrębie wiersza) liczby naturalne. Proszę napisać funkcję przepisującą wszystkie singletony (liczby występujące dokładnie raz) z tablicy  $T1$  do  $T2$ , tak aby liczby w tablicy  $T2$  były uporządkowane rosnąco. Pozostałe elementy tablicy  $T2$  powinny zawierać zera.

**Zadanie 104.** Dane są dwie tablice mogące pomieścić taką samą liczbę elementów:  $T1[N][N]$  i  $T2[M]$ , gdzie  $M=N*N$ . W każdym wierszu tablicy  $T1$  znajdują się uporządkowane niemalejąco (w obrębie wiersza) liczby naturalne. Proszę napisać funkcję przepisującą wszystkie liczby z tablicy  $T1$  do  $T2$ , tak aby liczby w tablicy  $T2$  były uporządkowane niemalejąco.

**Zadanie 105.** Dana jest tablica  $T[N][N]$  wypełniona liczbami naturalnymi. Proszę napisać funkcję, która w poszukuje w tablicy najdłuższego ciągu geometrycznego leżącego ukośnie w kierunku prawo-dół, liczącego co najmniej 3 elementy. Do funkcji należy przekazać tablicę. Funkcja powinna zwrócić informacje czy udało się znaleźć taki ciąg oraz długość tego ciągu.

**Zadanie 106.** Dana jest tablica  $T[N][N]$  wypełniona liczbami naturalnymi. Proszę napisać funkcję, która w poszukuje w tablicy kwadratu o liczbie pól będącej liczbą nieparzystą większą od 1, którego iloczyn 4 pól narożnych wynosi  $k$ . Do funkcji należy przekazać tablicę i wartość  $k$ . Funkcja powinna zwrócić informacje czy udało się znaleźć kwadrat oraz współrzędne (wiersz, kolumna) środka kwadratu.

**Zadanie 107.** Napisać funkcję która dla tablicy  $T[N][N]$ , wypełnionej liczbami całkowitymi, zwraca wartość `True` w przypadku, gdy w każdym wierszu i każdej kolumnie występuje co najmniej jedno 0 oraz wartość `False` w przeciwnym przypadku.

**Zadanie 108.** Dwie liczby naturalne są „przyjaciółkami” jeżeli zbiory cyfr z których zbudowane są liczby są identyczne. Na przykład: 123 i 321, 211 i 122, 35 3553. Dana jest tablica  $T[N][N]$  wypełniona liczbami naturalnymi. Proszę napisać funkcję, która dla tablicy  $T$  zwraca ile elementów tablicy sąsiaduje wyłącznie z przyjaciółkami

**Zadanie 109.** Dana jest tablica  $T[N][N][N]$ . Proszę napisać funkcję, do której przekazujemy tablicę wypełnioną liczbami większymi od zera. Funkcja powinna zwracać wartość `True`, jeżeli na wszystkich poziomach

tablicy liczba elementów sąsiadujących (w obrębia poziomym) z co najmniej 6 liczbami złożonymi jest jednokowa albo wartość False w przeciwnym przypadku.

**Zadanie 110.** Liczby naturalne  $a, b$  są komplementarne jeżeli ich suma jest liczbą pierwszą. Dana jest tablica  $T[N][N]$  wypełniona liczbami naturalnymi. Proszę napisać funkcję, która zeruje elementy nie posiadające liczby komplementarnej.

**Zadanie 111.** Dwie liczby naturalne są zgodne jeżeli w zapisie dwójkowym zawierają tę samą liczbę jedynek, np.  $22 = 101102$  i  $14 = 11102$ . Dane są tablice  $T1[N1][N1]$   $T2[N2][N2]$ , gdzie  $N2 \leq N1$ . Proszę napisać funkcję, która sprawdza czy istnieje takie położenie tablicy  $T1$  wewnątrz tablicy  $T2$ , przy którym liczba zgodnych elementów jest większa od 33%. Do funkcji należy przekazać tablicę  $T1$  i  $T2$ . Obie oryginalne tablice powinny pozostać nie zmieniane.

**Zadanie 112.** Dana jest tablica  $T[N][N]$ , wypełniona liczbami naturalnymi. Proszę napisać funkcję, która odpowiada na pytanie, czy w tablicy istnieje wiersz, w którym każda liczba zawiera co najmniej jedną cyfrę będącą liczbą pierwszą?

**Zadanie 113.** Dana jest tablica  $T[N][N]$ , wypełniona liczbami naturalnymi. Proszę napisać funkcję która odpowiada na pytanie, czy w tablicy każdy wiersz zawiera co najmniej jedną liczbą złożoną wyłącznie z cyfr będących liczbami pierwszymi?

**Zadanie 114.** Dana jest tablica  $T[N][N]$  wypełniona liczbami naturalnymi. Proszę napisać funkcję która zwraca wiersz i kolumnę dowolnego elementu, dla którego suma otaczających go elementów jest największa.

**Zadanie 115.** Dana jest tablica  $T[N][N]$  wypełniona liczbami całkowitymi. Proszę napisać funkcję, która wyszuka spójny podciąg elementów leżący poziomo lub pionowo o największej sumie. Maksymalna długość podciągu może wynosić 10 elementów. Do funkcji należy przekazać tablicę  $T$ , funkcja powinna zwrócić sumę maksymalnego podciągu.

**Zadanie 116.** Dana jest tablica  $T[N][N]$  wypełniona liczbami naturalnymi. Proszę napisać funkcję, która zwraca liczbę par elementów, o określonym iloczynie, takich że elementy są odległe o jeden ruch skoczka szachowego.

**Zadanie 117.** Dana jest plansza o wymiarach  $N \times N$  zawierająca wartości 0 i 1. Pola o wartości 1 zawierają pułapki. Skoczek musi dotrzeć z górnego wiersza planszy do dolnego. Każdy ruch skoczka musi go przybliżyć do dolnego wiersza. Proszę napisać program, który zwraca długość najkrótszej bezpiecznej drogi skoczka z wiersza górnego do wiersza dolnego.

**Zadanie 118.** Dana jest tablica  $T[N][N]$  wypełniona wartościami 0, 1. Każdy wiersz tablicy traktujemy jako liczbę zapisaną w systemie dwójkowym o długości  $N$  bitów. Stała  $N$  jest rzędu 1000. Proszę zaimplementować funkcję `distance(T)`, która dla takiej tablicy wyznaczy dwa wiersze, dla których różnica zawartych w wierszach liczb jest największa. Do funkcji należy przekazać tablicę, funkcja powinna zwrócić odległość pomiędzy znalezionymi wierszami. Można założyć, że żadne dwa wiersze nie zawierają identycznego ciągu cyfr.

**Zadanie 119.** Dana jest tablica  $T[N][N]$  (reprezentująca szachownicę) wypełniona liczbami całkowitymi. Proszę zaimplementować funkcję `chess(T)` która ustawia na szachownicy dwie wieże, tak aby suma liczb na „szachowanych” przez wieże polach była największa. Do funkcji należy przekazać tablicę, funkcja powinna zwrócić położenie wież w postaci krotki  $(row_1, col_1, row_2, col_2)$ .

Uwaga: zakładamy, że pola na których znajdują się wieże nie są szachowane.

Przykładowe wywołania funkcji:

```
print(chess([[4,0,2],[3,0,0],[6,5,3]])) # (0,1,1,0) suma=17
print(chess([[1,1,2,3],[-1,3,-1,4],[4,1,5,4],[5,0,3,6]])) # (2,3,3,1) suma=35
```

**Zadanie 120.** Dwie liczby naturalne są "wspólno-czynnikowe" jeżeli w swoich rozkładach na czynniki pierwsze mają dokładnie jeden wspólny czynnik. Na przykład: 24 i 21 albo 32 i 34. Dana jest tablica  $T[N][N]$  zawierająca liczby naturalne. Dwie liczby w tablicy sąsiadują ze sobą wtedy gdy leżą w tym samym wierszu i sąsiednich kolumnach albo leżą w tej samej kolumnie i sąsiednich wierszach. Proszę napisać funkcję `four(T)`, która zwraca ilość liczb sąsiadujących z 4 liczbami wspólnoczynnikowymi.

**Zadanie 121.** Na szachownicy o wymiarach  $N \times N$  wypełnionej liczbami naturalnymi większymi od 1 odbywają się wyścigi wież. Pierwsza wieża startuje z lewego górnego rogu i ma dotrzeć do prawego dolnego rogu szachownicy. Pierwsza wieża może wykonywać tylko ruchy w prawo lub w dół szachownicy. Druga wieża startuje z prawego górnego rogu i ma dotrzeć do lewego dolnego rogu szachownicy. Druga wieża może wykonywać tylko ruchy w lewo lub w dół szachownicy. Wygrywa wieża, która dotrze do mety w mniejszej liczbie ruchów. Wieże mogą wykonać ruch z jednego pola na drugie tylko wtedy, gdy liczby na obu polach są względnie pierwsze. Proszę napisać funkcję, która dla danej tablicy zwraca numer wieży, która wygra wyścig lub zero jeżeli wyścig będzie nierozstrzygnięty. Można założyć, że podczas wyścigu wieże nie spotkają się na jednym polu

**Zadanie 122.** W tablicy o rozmiarze  $N \times N$  wypełnionej liczbami naturalnymi umieszczono dokładnie jeden fragment ciągu Fibonacciego o długości co najmniej 3 elementów. Ciąg ten może leżeć w tablicy pionowo lub poziomo w kierunku rosnącym lub malejącym. Proszę napisać funkcję, która dla zadanej tablicy odszuka ten fragment i zwróci jego długość.

**Zadanie 123.** Dana jest tablica  $T[N][N]$  (reprezentująca szachownicę) wypełniona liczbami naturalnymi. W każdej kolumnie znajduje się dokładnie jedna wieża, której numer wiersza zawiera tablica `int w[N]`. Proszę napisać funkcję która wybiera do usunięcia z szachownicy dwie wieże, tak aby suma liczb na polach szachowanych przez pozostałe wieże była najmniejsza. Do funkcji należy przekazać tablice `t` i `w`, funkcja powinna zwrócić numery kolumn z których usunięto wieże. Uwaga - zakładamy, że wieża szachuje cały wiersz i kolumnę z wyłączeniem pola na którym stoi

**Zadanie 124.** Dana jest tablica `int t[N][N]` wypełniona przypadkowymi wartościami. Proszę napisać funkcję, która dla zmiennej typu `tablica` zwraca numer wiersza w którym występuje najdłuższy spójny fragment złożony z liczb o tej samej wartości. W przypadku kilku fragmentów o tej samej długości należy zwrócić pozycję pierwszego z nich.

**Zadanie 125.** Dwie liczby naturalne są 5-pokrewne, jeżeli po zapisaniu ich w systemie o podstawie 5, zbiory cyfr występujące w obu liczbach są identyczne. Na przykład:

$$21 = 41_{(5)} \text{ i } 34 = 114_{(5)}$$

$$27 = 102_{(5)} \text{ i } 51 = 201_{(5)}$$

Dana jest szachownica o rozmiarze  $N \times N$ , reprezentowana przez tablicę  $T$  wypełnioną liczbami naturalnymi. Sąsiedztwem danego pola na szachownicy są pola odległe o 1 lub 2 ruchy króla szachowego. Szczęśliwe pola to takie, które zawierają liczbę 5-pokrewną z liczbami na dokładnie 17 polach w swoim sąsiedztwie. Proszę napisać funkcję `luck17(T)`, która sprawdza czy w jakimkolwiek wierszu lub kolumnie znajduje się więcej niż jedno szczęśliwe pole. Do funkcji należy przekazać tablicę  $T$ , funkcja powinna zwrócić wartość `True` albo `False`. Tablica reprezentująca szachownicę nie może ulec zmianie.

**Zadanie 126.** Na szachownicy o rozmiarach  $N \times N$  reprezentowanej przez tablicę  $T[N][N]$  umieszczono pewną liczbę skoczków. Położenie skoczka w tablicy oznaczono liczbą 1, puste pola oznaczono liczbą 0. Część pustych pól na szachownicy jest szachowana przez znajdujące się na niej skoczki. Proszę zaproponować

funkcję `place(T)`, która znajdzie na szachownicy puste pole położone najbliżej środka szachownicy, takie że umieszczenie tam skoczka maksymalnie zwiększy liczbę szachowanych pustych pól. Do funkcji przekazujemy tablicę `T` zawierającą położenie skoczków. Funkcja powinna zwrócić pole (wiersz, kolumna), na którym należy umieścić skoczka. Odległość pomiędzy polami:  $(w_1, k_1)$  i  $(w_2, k_2)$  jest równa  $\max(\text{abs}(w_1 - w_2), \text{abs}(k_1 - k_2))$ .

**Zadanie 127.** Dwie liczby naturalne są "wspólno-czynnikowe" jeżeli w swoich rozkładach na czynniki pierwsze mają dokładnie jeden wspólny czynnik. Na przykład: 24 i 21 albo 32 i 34. Dana jest tablica  $T[N][N]$  zawierająca liczby naturalne. Dwie liczby w tablicy sąsiadują ze sobą wtedy, gdy leżą w tym samym wierszu i sąsiednich kolumnach, albo leżą w tej samej kolumnie i sąsiednich wierszach. Proszę napisać funkcję `four(T)`, która zwraca ilość liczb sąsiadujących z 4 liczbami wspólno-czynnikowymi.

**Zadanie 128.** Tablica  $T[N][N]$  zawiera liczby naturalne. Proszę napisać funkcję `select(T)`, która wybiera z tablicy dowolne  $N$  parami różnych elementów tak aby:

- żadne dwa wybrane elementy nie leżały w tej samej kolumnie ani wierszu,
- suma wybranych elementów powinna mieścić się w przedziale  $[2000, 2025]$ .

Funkcja powinna zwrócić wartość sumy wybranych elementów albo wartość `None`, jeżeli taki wybór nie jest możliwy.

**Zadanie 129.** Na szachownicy o wymiarach  $N \times N$  umieszczono pewną liczbę pionków. Położenie pionków opisuje lista  $[(w_0, k_0), (w_1, k_1), (w_2, k_2), \dots]$ . W lewym górnym rogu szachownicy (o współrzędnych 0, 0) znajduje się król, który musi dotrzeć do prawego dolnego rogu szachownicy. Król może wykonywać ruchy w prawo, w dół lub w górę szachownicy, nie może zbijać pionków ani wracać na pole, na którym już był. Proszę napisać funkcję `king(N, L)`, która zwróci maksymalną liczbę ruchów, jakie może wykonać król na drodze do celu. Do funkcji należy przekazać wyłącznie dwa parametry: rozmiar szachownicy  $N$  oraz listę  $L$  zawierającą położenia pionków. Jeżeli dotarcie do celu nie jest możliwe, funkcja powinna zwrócić wartość `None`.

## Zestaw 4: Struktury danych

**Zadanie 130.** Liczby wymierne są reprezentowane przez krotkę  $(l, m)$ . Gdzie:  $l$  - liczba całkowita oznaczająca licznik,  $m$  - liczba naturalna oznaczająca mianownik. Proszę napisać podstawowe operacje na ułamkach, m.in. dodawanie, odejmowanie, mnożenie, dzielenie, potęgowanie, skracanie, wypisywanie i wczytywanie.

**Zadanie 131.** Używając funkcji z poprzedniego zadania proszę napisać funkcję rozwiązującą układ 2 równań o 2 niewiadomych, w których współczynniki są liczbami wymiernymi

**Zadanie 132.** Na szachownicy o wymiarach 100 na 100 umieszczamy  $N$  hetmanów ( $N < 100$ ). Położenie hetmanów jest opisywane przez tablicę  $dane = [(w_1, k_1), (w_2, k_2), (w_3, k_3), \dots, (w_N, k_N)]$  Proszę napisać funkcję, która odpowiada na pytanie: czy żadne z dwa hetmany się nie szachują? Do funkcji należy przekazać położenie hetmanów.

**Zadanie 133.** Dana jest tablica zawierająca liczby wymierne. Proszę napisać funkcję, która policzy występujące w tablicy ciągi arytmetyczne (LA) i geometryczne (LG) o długości większej niż 2. Funkcja powinna zwrócić wartość 1 gdy  $LA > LG$ , wartość -1 gdy  $LA < LG$  oraz 0 gdy  $LA = LG$ .

**Zadanie 134.** Dany jest zbiór punktów leżących na płaszczyźnie opisany przy pomocy struktury  $dane = [(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_N, y_N)]$  Proszę napisać funkcję, która zwraca wartość True jeżeli zbiorze istnieją 4 punkty wyznaczające kwadrat o bokach równoległych do osi układu współrzędnych, a wewnątrz tego kwadratu nie ma żadnych innych punktów. Do funkcji należy przekazać strukturę opisującą położenie punktów.

**Zadanie 135.** Liczby zespolone są reprezentowane przez krotkę  $(re, im)$ . Gdzie:  $re$  - część rzeczywista liczby,  $im$  - część urojona liczby. Proszę napisać podstawowe operacje na liczbach zespolonych, m.in. dodawanie, odejmowanie, mnożenie, dzielenie, potęgowanie, wypisywanie i wczytywanie.

**Zadanie 136.** Używając funkcji z poprzedniego zadania proszę napisać funkcję rozwiązującą równanie kwadratowe o współczynnikach zespolonych.

**Zadanie 137.** Napis nazywamy wielokrotnym, jeżeli powstał przez  $n$ -krotne ( $n > 1$ ) powtórzenie innego napisu o długości co najmniej 1. Przykłady napisów wielokrotnych:  $ABCABCABC$ ,  $AAAA$ ,  $ABAABA$ . Dana jest tablica  $T[N]$  zawierająca napisy. Proszę napisać funkcję `multi(T)`, która zwraca długość najdłuższego napisu wielokrotnego występującego w tablicy  $T$  lub wartość 0, jeżeli takiego napisu nie ma w tablicy.

**Zadanie 138.** Dana jest tablica  $T[N][N]$  wypełniona wartościami 0, 1. Każdy wiersz tablicy traktujemy jako liczbę zapisaną w systemie dwójkowym o długości  $N$  bitów. Stała  $N$  jest rzędu  $10^3$ . Proszę zaimplementować funkcję `distance(T)`, która dla takiej tablicy wyznaczy dwa wiersze, dla których różnica zawartych w wierszach liczb jest największa. Do funkcji należy przekazać tablicę, funkcja powinna zwrócić odległość pomiędzy znalezionymi wierszami. Można założyć, że żadne dwa wiersze nie zawierają identycznego ciągu cyfr.

**Zadanie 139.** Proszę napisać funkcję która zamienia liczby wymierne reprezentowane jako rozwinięcia dziesiętne w postaci napisów na liczbę wymierną w postaci nieskracalnego ułamka jako pary licznik-mianownik. Na przykład: "0.25" na (1,4), "0.(6)" na (2,3), "0.1(6)" na (1,6), "0.(142857)" na (1,7)

**Zadanie 140.** Proszę napisać funkcję która dodaje dwie liczby wymierne reprezentowane jako rozwinięcia dziesiętne w postaci napisów na tak samo reprezentowaną liczbę wymierną. Na przykład suma "0.25" i "0.1(6)" daje "0.41(6)"

**Zadanie 141.** Liczby rzeczywiste są reprezentowane w postaci krotek  $(c, u)$ , gdzie  $c$  jest częścią całkowitą liczby, a  $u$  jest liczbą całkowitą utworzoną z cyfr po przecinku. Na przykład krotka (6, 23) reprezentuje liczbę 6.23. Proszę napisać następujące funkcje:

- Dodającą dwie liczby,
- Odejmującą dwie liczby,
- Mnożącą dwie liczby.

**Zadanie 142.** Proszę napisać program wczytujący równanie typu:  $abc+de = aff, abb-ab = acd, ab*ca = ade$ , w którym litery odpowiadają cyfrom. Program powinien wypisać wszystkie możliwe rozwiązania.

**Zadanie 143.** Dany jest ciąg  $N$  liczb naturalnych, z którego wybieramy spójny fragment o długości  $K$  ( $1 < K < N$ ). Pomiedzy wszystkie elementy wybranego fragmentu możemy wstawiać operatory dodawania albo mnożenia, tak aby powstało wyrażenie arytmetyczne. W powstałym wyrażeniu nie mogą wystąpić dwa jednakowe operatory obok siebie. Interesuje nas znalezienie takiego fragmentu ciągu, który pozwala zbudować wyrażenie o wartości będącej liczbą pierwszą. Proszę napisać funkcję  $find\_max(T)$ , która dla ciągu zawartego w tablicy  $T$ , wyznaczy wartość największej liczby pierwszej, jaką można znaleźć. Jeżeli taki podciąg nie istnieje, funkcja powinna zwrócić wartość zero.

Na przykład dla ciągu: 7,8,6,4,7,3 funkcja powinna zwrócić wartość 83

Możliwe podciągi dające liczby pierwsze to:

$$7 + 8 * 6 + 4 = 59$$

$$7 + 8 * 6 + 4 * 7 = 83$$

$$6 * 4 + 7 = 31$$

$$4 + 7 = 11$$

## Zestaw 5: Rekurencja

**Zadanie 144.** Korzystając z zależności:  $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$  proszę napisać funkcję obliczającą wartość symbolu Newtona dla argumentów  $n$  i  $k$ .

**Zadanie 145.** Proszę napisać funkcję, która jako argument przyjmuje liczbę całkowitą i wypisuje wszystkie co najmniej dwucyfrowe liczby pierwsze, powstałe poprzez wykreślenie z liczby pierwotnej co najmniej jednej cyfry.

**Zadanie 146.** "Waga" liczby jest określona jako liczba różnych czynników pierwszych liczby. Na przykład  $waga(1)=0$ ,  $waga(2)=1$ ,  $waga(6)=2$ ,  $waga(30)=3$ ,  $waga(64)=1$ . Dana jest tablica  $T[N]$  zawierająca liczby naturalne. Proszę napisać funkcję, która sprawdza czy można elementy tablicy podzielić na 3 podzbiory o równych wagach. Do funkcji należy przekazać wyłącznie tablicę, funkcja powinna zwrócić wartość typu Bool.

**Zadanie 147.** Szachownica jest reprezentowana przez tablicę  $T[8][8]$  wypełnioną liczbami naturalnymi zawierającymi koszt przebywania na danym polu szachownicy. Król szachowy znajduje się w wierszu 0 i kolumnie  $k$ . Król musi w dokładnie 7 ruchach dotrzeć do wiersza 7. Proszę napisać funkcję, która wyznaczy minimalny koszt przejścia króla. Do funkcji należy przekazać tablicę  $t$  oraz startową kolumnę  $k$ . Koszt przebywania na polu startowym i ostatnim także wliczamy do kosztu przejścia.

**Zadanie 148.** Problem skoczka szachowego. Proszę napisać funkcję, która wypełnia pola szachownicy o wymiarach  $N \times N$  ruchem skoczka szachowego.

**Zadanie 149.** Dany jest ciąg zer i jedynek zapisany w tablicy  $T[N]$ . Proszę napisać funkcję, która odpowiada na pytanie czy jest możliwe pocięcie ciągu na kawałki z których każdy reprezentuje liczbę pierwszą. Długość każdego z kawałków nie może przekraczać 30. Na przykład dla ciągu 111011 jest to możliwe, a dla ciągu 110100 nie jest możliwe.

**Zadanie 150.** Dana jest tablica  $T[N]$ . Proszę napisać funkcję, która znajdzie niepusty, najmniejszy (w sensie liczebności) podzbiór elementów tablicy, dla którego suma elementów jest równa sumie indeksów tych elementów. Do funkcji należy przekazać tablicę, funkcja powinna zwrócić sumę elementów znalezionego podzbioru. Na przykład dla tablicy: [ 1,7,3,5,11,2 ] rozwiązaniem jest liczba 10, są to elementy o indeksach 2,3,i 5.

**Zadanie 151.** Liczba dwu-trzy-piątkowa w rozkładzie na czynniki pierwsze nie posiada innych czynników niż 2,3,5. Jedynek też jest taką liczbą. Proszę napisać funkcję rekurencyjną, wypisującą liczby znajdujące się w przedziale od 1 do  $N$  włącznie.

**Zadanie 152.** Dany jest zestaw odważników  $T[N]$ . Napisać funkcję, która sprawdza czy jest możliwe odważenie określonej masy. Odważniki można umieszczać tylko na jednej szalce.

**Zadanie 153.** Poprzednie zadanie, ale odważniki można umieszczać na obu szalkach.

**Zadanie 154.** Poprzednie zadanie. Program powinien wypisywać wybrane odważniki.

**Zadanie 155.** Rekurencyjne obliczanie wyznacznika z macierzy (treść oczywista)

**Zadanie 156.** Dana jest tablica  $T[N]$ . Proszę napisać program zliczający liczbę "enek" o określonym iloczynie.

**Zadanie 157.** Proszę zmodyfikować poprzedni program aby wypisywał znalezione  $n$ -ki.

**Zadanie 158.** Napisać program wypisujący wszystkie możliwe podziały liczby naturalnej na sumę składników. Na przykład dla liczby 4 są to: 1+3, 1+1+2, 1+1+1+1, 2+2.

**Zadanie 159.** Problem wież w Hanoi (treść oczywista)

**Zadanie 160.** Problem 8 Hetmanów (treść oczywista)

**Zadanie 161.** Wyrazy budowane są z liter a..z. Dwa wyrazy "ważą" tyle samo jeżeli: mają tę samą liczbę samogłosek oraz sumy kodów ascii liter z których są zbudowane są identyczne, na przykład "ula" → 117, 108, 97 oraz "exe" → 101, 120, 101. Proszę napisać funkcję `wyraz(s1,s2)`, która sprawdza czy jest możliwe zbudowanie wyrazu z podzbioru liter zawartych w s2 ważącego tyle co wyraz s1. Dodatkowo funkcja powinna wypisać znaleziony wyraz.

**Zadanie 162.** Dane są dwie liczby naturalne z których budujemy trzecią liczbę. W budowanej liczbie muszą wystąpić wszystkie cyfry występujące w liczbach wejściowych. Wzajemna kolejność cyfr każdej z liczb wejściowych musi być zachowana. Na przykład mając liczby 123 i 75 możemy zbudować liczby 12375, 17523, 75123, 17253, itd. Proszę napisać funkcję która wyznaczy ile liczb pierwszych można zbudować z dwóch zadanych liczb.

**Zadanie 163.** W szachownicy o wymiarach 8x8 każdemu z pól przypisano liczbę naturalną. Na ruchy króla nałożono dwa ograniczenia: król może przesunąć się na jedno z 8 sąsiednich pól jeżeli ostatnia cyfra liczby na polu na którym stoi jest mniejsza od pierwszej cyfry liczby pola docelowego, oraz w drodze do obranego celu (np. narożnika) król nie może wykonać ruchu, który powoduje oddalenie go od celu. Dana jest globalna tablica `T[8][8]` wypełniona liczbami naturalnymi reprezentująca szachownicę. Lewy górny narożnik ma współrzędne `w=0` i `k=0`. Proszę napisać funkcję sprawdzającą czy król może dostać się z pola `w,k` do prawego dolnego narożnika.

**Zadanie 164.** Zadanie jak powyżej. Funkcja sprawdzająca czy król może dostać się z pola `w,k` do któregośkolwiek z narożników.

**Zadanie 165.** Zadanie jak powyżej. Funkcja powinna dostarczyć drogę króla w postaci tablicy zawierającej kierunki (liczby od 0 do 7) poszczególnych ruchów króla do wybranego celu.

**Zadanie 166.** Tablica `T[8][8]` zawiera liczby naturalne. Proszę napisać funkcję, która sprawdza czy można wybrać z tablicy niepusty podzbiór o zadanej sumie. Warunkiem dodatkowym jest aby żadne dwa wybrane elementy nie leżały w tej samej kolumnie ani wierszu. Do funkcji należy przekazać wyłącznie tablicę oraz wartość sumy, funkcja powinna zwrócić wartość typu `bool`.

**Zadanie 167.** Dana jest tablica `T[N]` zawierająca liczby naturalne. Po tablicy możemy przemieszczać się według następującej zasady: z pola o indeksie `i` możemy przeskoczyć na pole o indeksie `i+k` jeżeli `k` jest czynnikiem pierwszym liczby `t[i]` mniejszym od `t[i]`. Proszę napisać funkcję, która zwraca informację czy jest możliwe przejście z pola o indeksie 0 na pole o indeksie `N-1`. Funkcja powinna zwrócić liczbę wykonanych skoków lub wartość -1 jeżeli powyższe przejście nie jest możliwe.

**Zadanie 168.** Dana jest tablica `T[N]` zawierająca oporności `N` rezystorów wyrażonych całkowitą liczbą  $k\Omega$ . Proszę napisać funkcję, która sprawdza czy jest możliwe uzyskanie wypadkowej rezystancji `R` (równej całkowitej liczbie  $k\Omega$ ) łącząc dowolnie 3 wybrane rezystory.

**Zadanie 169.** Tablica `T = [(x1, y1), (x1, y1), ...]` zawiera położenia `N` punktów o współrzędnych opisanych wartościami typu `float`. Proszę napisać funkcję, która zwróci najmniejszą odległość między środkami ciężkości 2 niepustych podzbiorów tego zbioru.

**Zadanie 170.** Do budowy liczby naturalnej reprezentowanej w systemie dwójkowym możemy użyć  $A$  cyfr 1 oraz  $B$  cyfr 0, gdzie  $A, B > 0$ . Proszę napisać funkcję, która dla zadanych parametrów  $A$  i  $B$  zwraca ilość wszystkich możliwych do zbudowania liczb, takich że pierwsza cyfra w systemie dwójkowym (najstarszy bit) jest równa 1, a zbudowana liczba jest złożona. Na przykład dla  $A=2$ ,  $B=3$  ilość liczb wynosi 3, są to  $10010_{(2)}$ ,  $10100_{(2)}$ ,  $11000_{(2)}$

**Zadanie 171.** Kwadrat jest opisywany czwórką liczb całkowitych  $(x_1, x_2, y_1, y_2)$ , gdzie  $x_1, x_2, y_1, y_2$  oznaczają proste ograniczające kwadrat  $x_1 < x_2$ ,  $y_1 < y_2$ . Dana jest tablica  $T$  zawierająca opisy  $N$  kwadratów. Proszę napisać funkcję, która zwraca wartość logiczną `True`, jeśli danej tablicy można znaleźć 13 nienachodzących na siebie kwadratów, których suma pól jest równa 2012 i `False` w przeciwnym przypadku.

**Zadanie 172.** Dany jest zbiór  $N$  liczb naturalnych umieszczony w tablicy  $T[N]$ . Proszę napisać funkcję, która zwraca informację, czy jest możliwy podział zbioru  $N$  liczb na trzy podzbiory, tak aby w każdym podzbiorze, łączna liczba jedynek użyta do zapisu elementów tego podzbioru w systemie dwójkowym była jednakowa. Na przykład:  $[2, 3, 5, 7, 15] \rightarrow \text{true}$ , bo podzbiory  $\{2, 7\}$   $\{3, 5\}$   $\{15\}$  wymagają użycia 4 jedynek,  $[5, 7, 15] \rightarrow \text{false}$ , podział nie istnieje.

**Zadanie 173.** Punkt leżący w przestrzeni jest opisywany trójką liczb typu float  $(x, y, z)$ . Tablica  $T[N]$  zawiera współrzędne  $N$  punktów leżących w przestrzeni. Punkty posiadają jednostkową masę. Proszę napisać funkcję, która sprawdza czy istnieje podzbiór punktów liczący co najmniej 3 punkty, którego środek ciężkości leży w odległości nie większej niż  $r$  od początku układu współrzędnych. Do funkcji należy przekazać tablicę  $T$  oraz promień  $r$ , funkcja powinna zwrócić wartość typu `bool`.

**Zadanie 174.** Punkt leżący na płaszczyźnie jest opisywany parą liczb typu float  $(x, y)$ . Tablica  $T[N]$  zawiera współrzędne  $N$  punktów leżących na płaszczyźnie. Punkty posiadają jednostkową masę. Proszę napisać funkcję, która sprawdza czy istnieje niepusty podzbiór  $n$  punktów, gdzie  $n \geq k$  oraz  $n$  jest wielokrotnością liczby 3, którego środek ciężkości leży w odległości mniejszej niż  $r$  od początku układu współrzędnych. Do funkcji należy przekazać dokładnie 3 parametry: tablicę  $t$ , promień  $r$ , oraz ograniczenie  $k$ , funkcja powinna zwrócić wartość typu `bool`.

**Zadanie 175.** Proszę napisać funkcję, która jako parametr otrzymuje liczbę naturalną i zwraca sumę iloczynów elementów wszystkich niepustych podzbiorów zbioru dzielników pierwszych tej liczby. Można założyć, że liczba dzielników pierwszych nie przekracza 20, zatem w pierwszym etapie funkcja powinna wpisać podzielniki do tablicy pomocniczej. Przykład:  $60 \rightarrow [2, 3, 5] \rightarrow 2+3+5+2*3+2*5+3*5+2*3*5 = 71$

**Zadanie 176.** Dana jest tablica  $T[N]$  zawierająca liczby naturalne. Proszę napisać funkcję, która odpowiada na pytanie, czy spośród (niekoniecznie wszystkich) elementów tablicy można utworzyć dwa podzbiory o jednakowej sumie elementów, tak aby suma mocy obu podzbiorów wynosiła  $k$ . Do funkcji należy przekazać wyłącznie tablicę  $T$  oraz liczbę naturalną  $k$ , funkcja powinna zwrócić wartość typu `bool`.

**Zadanie 177.** Dwie osoby otrzymały ten sam ciąg liter. Każda osoba pocięła go na kawałki i pomieszała. Proszę napisać program, który otrzymując dwa zbiory kawałków odtwarza napis początkowy jeżeli odtworzenie to jest jednoznaczne lub stwierdza brak możliwości jednoznacznego odtworzenia napisu. Na przykład dla zbiorów (1)  $ab, cde, cfed, fab$  (2)  $abc, abc, def, fed$  odtworzony napis to:  $abcdefabcfed$

**Zadanie 178.** Dane jest słowo składające się z liter alfabetu angielskiego. Słowo to tnijemy na co najmniej dwa kawałki, tak aby każdy kawałek zawierał dokładnie jedną samogłoskę. Proszę napisać funkcję, która zwraca liczbę sposobów pocięcia słowa na kawałki.

**Zadanie 179.** Dana jest liczba naturalna  $N$ . Proszę zaimplementować funkcję `divide(N)`, która sprawdza czy jest możliwe pocięcie liczby  $N$  na kawałki, tak aby każdy z kawałków był liczbą pierwszą oraz liczba kawałków też była liczbą pierwszą. Funkcja powinna zwracać wartość logiczną. Na przykład: `divide(2347)=True`,

podział na 23 i 47, natomiast `divide(2255)=False`.

Przykładowe wywołania funkcji:

```
print(divide(273)) # True, podział 2|7|3
print(divide(22222)) # True, podział 2|2|2|2|2
print(divide(23672)) # True, podział 23|67|2
print(divide(2222)) # False
print(divide(21722)) # False
```

**Zadanie 180.** Dwie liczby naturalne są "wspólno-czynnikowe" jeżeli w swoich rozkładach na czynniki pierwsze mają dokładnie jeden wspólny czynnik. Na przykład: 24 i 21 albo 32 i 34. Dana jest tablica  $T[N][N]$  zawierająca liczby naturalne. Dwie liczby w tablicy sąsiadują ze sobą wtedy gdy leżą w tym samym wierszu i sąsiednich kolumnach albo leżą w tej samej kolumnie i sąsiednich wierszach. Proszę napisać funkcję `four(T)`, która zwraca ilość liczb sąsiadujących z 4 liczbami wspólnoczynnikowymi.

**Zadanie 181.** Dana jest funkcja Ackermana, określona na zbiorze liczb całkowitych nieujemnych, dana wzorem rekurencyjnym:

```
def f(a,b):
    if a==0: return b+1
    if b==0: return f(a-1,1)
    return f(a-1,f(a,b-1))
```

Proszę napisać funkcję w wersji iteracyjnej.

**Zadanie 182.** Dana jest funkcja rozwiązująca problem wież Hanoi:

```
def hanoi(n,a,b,c):
    if n>0:
        hanoi(n-1,a,c,b)
        print(a,'->',c)
        hanoi(n-1,b,a,c)
```

- Usunąć rekurencję zastępując ją stosem,
- Zaimplementować algorytm bez użycia stosu ani rekurencji (wikipedia).

**Zadanie 183.** Proszę napisać rekurencyjną funkcję obliczającą  $n$ -ty wyraz ciągu Fibonacciego ale tak aby wewnątrz funkcji było tylko jedno odwołanie rekurencyjne.

**Zadanie 184.** Dane są trzy operacje: A,B,C przekształcające liczbę naturalną  $x$  na inną liczbę naturalną.

- A zamienia liczbę  $x$  na jej rewers i dodaje 1. Na przykład:  $123 \rightarrow 322$ ,  $230 \rightarrow 33$ .
- B zamienia liczbę  $x$  na najmniejszą liczbę pierwszą większą od  $x$ . Na przykład  $13 \rightarrow 17$ ,  $20 \rightarrow 23$ .
- C oblicza odwrotność liczby  $x$  i zwraca liczbę złożoną z 3 cyfr rozwinięcia dziesiętnego odwrotności, po pominięciu początkowych zer. Na przykład:  $6 \rightarrow 166$  bo  $1/6 = 0.16666\dots$ ,  $10 \rightarrow 100$  bo  $1/10 = 0.1000\dots$ ,  $12 \rightarrow 833$  bo  $1/12 = 0.083333\dots$ ,  $997 \rightarrow 100$  bo  $1/997 = 0.001003009\dots$

Proszę napisać funkcję `cykl(x)`, która dla liczby naturalnej  $x$  zwraca minimalną sekwencję operacji, których wykonanie powraca do liczby początkowej  $x$ . Do funkcji należy przekazać wyłącznie wartość początkową  $x$ , funkcja powinna zwrócić napis złożony z liter A,B,C. Jeżeli nie jest możliwy powrót do wartości początkowej w mniej niż 10 operacjach, funkcja powinna zwrócić napis pusty. Można założyć, że  $x \leq 10^6$ .

Przykłady:

```
cykl(3)="BCA", 3 → 5 → 200 → 3,  
cykl(35)="BBBA", 35 → 37 → 41 → 43 → 35,  
cykl(45)="CABCA", 45 → 222 → 223 → 227 → 440 → 45,  
cykl(51)="", cykl krótszy niż 10 nie istnieje.
```

**Zadanie 185.** Dane jest  $N$  szkatulek, każda mogąca pomieścić maksymalnie 20 sztabek złota. Możemy przenosić sztabki złota pomiędzy szkatułkami, jednak liczba sztabek w żadnej szkatulce nie może zmienić się o więcej niż 2 sztabki. Dana jest tablica  $T[N]$  zawierająca liczby sztabek w poszczególnych szkatułkach. Proszę napisać funkcję `gold(T)`, która sprawdza, czy jest możliwe takie przeniesienie części sztabek pomiędzy szkatułkami, tak aby w liczba sztabek w każdej szkatulce była liczbą pierwszą. Funkcja powinna zwrócić wartość *True* albo *False*.

Przykłady:

```
gold([4, 18, 15, 14, 14, 6]) powinna zwrócić True, nowe wartości [2, 19, 17, 13, 13, 7],  
gold([3, 10, 11, 18, 16, 16]) powinna zwrócić False
```

**Zadanie 186.** Ciąg liczb naturalnych jest zawarty w tablicy  $T$ . Proszę napisać funkcję `sequence(T)` która sprawdza, czy jest możliwe pocięcie ciągu na kawałki w taki sposób, aby sumy elementów w kolejnych kawałkach stanowiły rosnący ciąg arytmetyczny. Funkcja powinna zwrócić wartość pierwszego wyrazu tego ciągu oraz jego różnicę. Jeżeli utworzenie tego ciągu jest niemożliwe, funkcja powinna zwrócić wartość *None*.

Przykład: Dla ciągu: 2,3,5,3,3,7,3,7,23,11,13,5 funkcja powinna zwrócić wartości 5,6 powstały ciąg to 5,11,17,23,29.

## Zestaw 6: Struktury odsyłaczowe liniowe

**Zadanie 187.** wypisywanie elementów łańcucha

**Zadanie 188.** Zliczanie elementów łańcucha

**Zadanie 189.** Proszę zaimplementować zbiór mnogościowy liczb naturalnych korzystając ze struktury listy odsyłaczowej.

- czy element należy do zbioru
- wstawienie elementu do zbioru
- usunięcie elementu ze zbioru

**Zadanie 190.** Zastosowanie listy odsyłaczowej do implementacji tablicy rzadkiej. Proszę napisać trzy funkcje:

- inicjalizującą tablicę,
- zwracającą wartość elementu o indeksie  $n$ ,
- podstawiającą wartość *value* pod indeks  $n$ .

**Zadanie 191.** Proszę napisać funkcję scalającą dwie posortowane listy w jedną posortowaną listę. Do funkcji należy przekazać wskazania na pierwsze elementy obu list, funkcja powinna zwrócić wskazanie do scalonej listy. Zadanie należy wykonać jako funkcję iteracyjną, a następnie jako funkcję rekurencyjną.

**Zadanie 192.** Proszę napisać funkcję, która dla podanej listy odsyłaczowej odwraca kolejność jej elementów.

**Zadanie 193.** Proszę napisać funkcję, która rozdziela elementy listy odsyłaczowej do 10 list, według ostatniej cyfry pola *val*. W drugim kroku powstałe listy należy połączyć w jedną listę odsyłaczową, która jest posortowana niemalejąco według ostatniej cyfry pola *val*.

**Zadanie 194.** Proszę napisać funkcję wstawiającą na koniec listy nowy element. Do funkcji należy przekazać wskazanie na pierwszy element listy oraz wstawianą wartość.

**Zadanie 195.** Proszę napisać funkcję usuwającą ostatni element listy. Do funkcji należy przekazać wskazanie na pierwszy element listy.

**Zadanie 196.** Dana jest niepusta lista, proszę napisać funkcję usuwającą co drugi element listy. Do funkcji należy przekazać wskazanie na pierwszy element listy.

**Zadanie 197.** Dana jest niepusta lista reprezentująca liczbę naturalną. Kolejne elementy listy przechowują kolejne cyfry. Proszę napisać funkcję zwiększającą taką liczbę o 1.

**Zadanie 198.** Liczby naturalne reprezentowane jak poprzednim zadaniu. Proszę napisać funkcję dodającą dwie takie liczby. W wyniku dodawania dwóch liczb powinna powstać nowa lista.

**Zadanie 199.** Lista zawiera niepowtarzające się elementy. Proszę napisać funkcję do której przekazujemy wskaźnik na początek listy oraz wartość klucza. Jeżeli element o takim kluczu występuje w liście należy go usunąć z listy. Jeżeli elementu o zadanym kluczu brak jest w liście należy element o takim kluczu wstawić do listy.

**Zadanie 200.** Zbiór mnogościowy zawierający napisy jest reprezentowany w postaci jednokierunkowej listy. Napisy w łańcuchu są uporządkowane leksykograficznie. Proszę napisać stosowne definicje typów oraz funkcję dodającą napis do zbioru. Do funkcji należy przekazać wskaźnik do listy oraz wstawiany napis, funkcja powinna zwrócić wartość logiczną wskazującą, czy w wyniku operacji moc zbioru uległa zmianie.

**Zadanie 201.** Proszę napisać funkcję, otrzymującą jako parametr wskaźnik na pierwszy element listy o wartościach typu `int`, usuwającą wszystkie elementy, których wartość jest mniejsza od wartości bezpośrednio poprzedzających je elementów.

**Zadanie 202.** Proszę napisać funkcję, otrzymującą jako parametr wskaźnik na pierwszy element listy o wartościach typu `int`, usuwającą wszystkie elementy, których wartość dzieli bez reszty wartość bezpośrednio następujących po nich elementów.

**Zadanie 203.** Proszę napisać funkcję, która otrzymując jako parametr wskazujący na początek listy jednokierunkowej, usuwa z niej wszystkie elementy, w których wartość klucza w zapisie trójkowym ma większą ilość jedynek niż dwójek.

**Zadanie 204.** Proszę napisać funkcję, która otrzymując jako parametr wskazujący na początek listy jednokierunkowej, przenosi na początek listy te z nich, które mają parzystą ilość piątek w zapisie ósemkowym.

**Zadanie 205.** Proszę napisać funkcję, która otrzymując jako parametr wskazujący na początek listy dwukierunkowej, usuwa z niej wszystkie elementy, w których wartość klucza w zapisie binarnym ma nieparzystą ilość jedynek.

**Zadanie 206.** Proszę napisać funkcję, która pozostawia w liście wyłącznie elementy unikalne. Do funkcji należy przekazać wskazanie na pierwszy element listy.

**Zadanie 207.** Elementy w liście są uporządkowane według wartości klucza. Proszę napisać funkcję usuwającą z listy elementy o nieunikalnym kluczu. Do funkcji przekazujemy wskazanie na pierwszy element listy, funkcja powinna zwrócić liczbę usuniętych elementów.

**Zadanie 208.** Dana jest lista zawierająca ciąg obustronnie domkniętych przedziałów. Krańce przedziałów określa uporządkowana para liczb całkowitych. Proszę napisać stosowne deklaracje oraz funkcję redukującą liczbę elementów listy. Na przykład lista: `[15,19] [2,5] [7,11] [8,12] [5,6] [13,17]` powinien zostać zredukowany do listy: `[13,19] [2,6] [7,12]`

**Zadanie 209.** Kolejne elementy listy o zwiększającej się wartości pola `val` nazywamy podlistą rosnącą. Proszę napisać funkcję, która usuwa z listy wejściowej najdłuższą podlistę rosnącą. Warunkiem usunięcia jest istnienie w liście dokładnie jednej najdłuższej podlisty rosnącej.

**Zadanie 210.** Dana jest lista, która może być zakończona jest cyklem. Napisać funkcję, która sprawdzi ten fakt.

**Zadanie 211.** Dana jest lista, która zakończona jest cyklem. Napisać funkcję, która zwraca liczbę elementów w cyklu.

**Zadanie 212.** Dana jest lista, która zakończona jest cyklem. Napisać funkcję, która zwraca liczbę elementów przed cyklem.

**Zadanie 213.** Dana jest lista, która zakończona jest cyklem. Napisać funkcję, która zwraca wskaźnik do ostatniego elementu przed cyklem.

**Zadanie 215.** Proszę napisać funkcję scalającą dwie posortowane listy w jedną posortowaną listę. Do funkcji należy przekazać wskazania na pierwsze elementy obu list, funkcja powinna zwrócić wskazanie do scalonej listy. - funkcja iteracyjna, - funkcja rekurencyjna.

**Zadanie 217.** Dwie listy zawierają niepowtarzające się (w obrębie listy) liczby naturalne. W obu listach liczby są posortowane rosnąco. Proszę napisać funkcję usuwającą z każdej listy liczby nie występujące w drugiej. Do funkcji należy przekazać wskazania na obie listy, funkcja powinna zwrócić łączną liczbę usuniętych elementów.

$$2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 11$$

**Zadanie 220.** Lista reprezentuje wielomian o współczynnikach całkowitych. Elementy w liście ułożone są według rosnących potęg. Proszę napisać funkcję obliczającą różnicę dwóch dowolnych wielomianów. Wielomiany reprezentowane są przez wyżej opisane listy. Procedura powinna zwracać wskaźnik do nowo utworzonej listy reprezentującej wielomian wynikowy. Listy wejściowe powinny pozostać niezmienione.

??bartek??leszek??marek??ola??zosia?? ??? Proszę napisać stosowne definicje typów oraz funkcję wstawiającą do listy napis z zachowaniem zasady poprzedzania. Do funkcji należy przekazać wskaźnik do listy oraz wstawiany napis, funkcja powinna zwrócić wartość logiczną wskazującą, czy udało się wstawić napis do listy. Po wstawieniu elementu wskaźnik do listy powinien wskazywać na nowo wstawiony element.

```
..def init(self, val, next=None):
```

```
....self.val = val
....self.next = next
```

Lista zawierała wartości stanowiące kolejne wyrazy ciągu arytmetycznego. Z wnętrza listy usunięto pewną liczbę elementów. Proszę napisać funkcję `repair(p)`, (`p` wskazuje na pierwszy element listy) która uzupełnia listę elementami, tak aby ponownie zawierała kolejne wyrazy ciągu arytmetycznego. Funkcja powinna zwrócić liczbę wstawionych elementów. Można założyć, że lista wejściowa liczy więcej niż 2 elementy.

**Zadanie 224.** Dana jest definicja klasy, której obiekty stanowią elementy listy odsyłaczowej:

```
class Node
..def init(self, val):
....self.val = val
....self.next = None
```

Zbiór mnogościowy liczb naturalnych reprezentowany jest przez listę o uporządkowanych rosnąco elementach. Proszę napisać funkcję `iloczyn(z1, z2, z3)`, która przekształca 3 listy (zbiory) `z1, z2, z3` w jedną listę (zbiór) zawierającą elementy będące częścią wspólną zbiorów `z1, z2, z3`. Funkcja powinna zwrócić wskazanie do listy wynikowej.

**Zadanie 225.** Dana jest definicja klasy, której obiekty stanowią elementy listy odsyłaczowej:

```
class Node:
..def __init__(self, val, next=None):
....self.val = val
....self.next = next
```

Lista zawiera wartości będące liczbami naturalnymi. Proszę napisać funkcję `repair(p)`, (`p` wskazuje na pierwszy element listy) która przekształca listę tak aby liczby parzyste znalazły się na końcu listy. Funkcja powinna zwrócić wskazanie na przekształconą listę. Można założyć, że lista wejściowa liczy więcej niż 2 elementy.

**Zadanie 226.** A teraz proszę rozwiązać wszystkie powyższe zadania przy założeniu że listy zawierają wartownika.

**Zadanie 227.** Dana jest definicja klasy, której obiekty stanowią elementy listy odsyłaczowej: `class Node: ..def init(self, val, next=None): ....self.val = val ....self.next = next` Lista zawierała wartości stanowiące kolejne wyrazy ciągu arytmetycznego. Z wnętrza listy usunięto pewną liczbę elementów. Proszę napisać funkcję `repair(p)`, (`p` wskazuje na pierwszy element listy) która uzupełnia listę elementami, tak aby ponownie zawierała kolejne wyrazy ciągu arytmetycznego. Funkcja powinna zwrócić liczbę wstawionych elementów. Komentarz: Można założyć, że lista wejściowa liczy więcej niż 2 elementy

**Zadanie 228.** Zadanie z siatką.

**Zadanie 229.** Listy odsyłaczowe budowane są z obiektów zawierających dwa pola: `val` przechowujące liczbę naturalną oraz pole `next` przechowujące wskaźnik do kolejnego elementu. Dane były dwie niepuste listy odsyłaczowe równej długości, każda zawierała rosnący ciąg arytmetyczny zaczynający się od wartości 1. Nieudane scalanie tych list spowodowało powstanie jednej listy o całkowicie pomieszanej kolejności elementów. Proszę napisać funkcję `fix(p)`, która z takiej listy odtwarza dwie listy sprzed nieudanego scalania. Do funkcji przekazujemy wskaźnik na scaloną listę, funkcja powinna zwrócić dwa wskaźniki na odtworzone listy.

Na przykład dla listy:

$9 \rightarrow 7 \rightarrow 11 \rightarrow 3 \rightarrow 26 \rightarrow 36 \rightarrow 13 \rightarrow 15 \rightarrow 31 \rightarrow 21 \rightarrow 6 \rightarrow 16 \rightarrow 1 \rightarrow 11 \rightarrow 1 \rightarrow 5$

Powinny zostać odtworzone listy:

$1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 11 \rightarrow 13 \rightarrow 15$

$1 \rightarrow 6 \rightarrow 11 \rightarrow 16 \rightarrow 21 \rightarrow 26 \rightarrow 31 \rightarrow 36$

**Zadanie 230.** Dana jest lista, zbudowana z elementów zawierających pola *val* i *next*, której węzły przechowują parami różne liczby naturalne. Proszę napisać funkcję `sort(p)`, która porządkuje elementy listy wskazywanej przez wskaźnik *p*, według niemalejącej liczby jedynek w kluczu, zapisanym w systemie o podstawie 2, a w przypadku identycznej liczby jedynek liczba mniejsza powinna poprzedzać większą. Funkcja powinna zwrócić wskazanie do uporządkowanej listy.

Na przykład dla listy zawierającej elementy: 31 7 13 5 11 3 2 1 32

uporządkowana lista wygląda następująco: 1 2 32 3 5 7 11 13 31

**Zadanie 231.** Lista odsyłaczowa zbudowana jest z obiektów zawierających dwa pola: *val* przechowujące liczbę naturalną oraz pole *next* przechowujące wskaźnik do kolejnego elementu. Dane była niepusta lista odsyłaczowa, która zawierała rosnący ciąg wartości *val*. Przez pomyłkę spójny fragment tej listy liczący 6 elementów został przeniesiony w inne miejsce w liście. Proszę napisać funkcję `fix(p)`, która naprawia taką listę, tak aby przywrócić rosnący porządek elementów. Do funkcji przekazujemy wskaźnik na listę, funkcja powinna zwrócić dwa wskaźniki na naprawioną listę.

Przykładowe "zepsute" listy:

$2 \rightarrow 3 \rightarrow 23 \rightarrow 29 \rightarrow 5 \rightarrow 7 \rightarrow 11 \rightarrow 13 \rightarrow 17 \rightarrow 19 \rightarrow 31 \rightarrow 37 \rightarrow 41 \rightarrow 43 \rightarrow 47 \rightarrow 53$

przeniesiony fragment 5 do 19

$17 \rightarrow 19 \rightarrow 23 \rightarrow 29 \rightarrow 31 \rightarrow 37 \rightarrow 41 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 11 \rightarrow 13 \rightarrow 43 \rightarrow 47 \rightarrow 53$

przeniesiony fragment 2 do 13

Powinny zostać przekształcone do listy:

$2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 11 \rightarrow 13 \rightarrow 17 \rightarrow 19 \rightarrow 23 \rightarrow 29 \rightarrow 31 \rightarrow 37 \rightarrow 41 \rightarrow 43 \rightarrow 47 \rightarrow 53$

**Zadanie 232.** Dana jest definicja klasy, której obiekty stanowią elementy listy odsyłaczowej:

```
class Node:
    ..def init(self, val, next=None):
        ....self.val = val
        ....self.next = next
```

Lista zawierała liczby naturalne stanowiące kolejne wyrazy ciągu  $A_n$  spełniającego warunek  $A_n = A_{n-1} + A_{n-2}$ . Z wnętrza listy usunięto pewną liczbę elementów, ale:

1. Pierwszy i ostatni element ciągu pozostały w liście,
2. Co najmniej trzy kolejne elementy ciągu znajdują się w liście.

Proszę napisać funkcję `repair(p)`, (*p* wskazuje na pierwszy element listy) która uzupełnia listę elementami, tak aby ponownie zawierała kolejne wyrazy ciągu  $A_n$ . Funkcja powinna zwrócić liczbę wstawionych elementów.

**Zadanie 233.** Dana jest definicja klasy, której obiekty stanowią elementy listy odsyłaczowej:

```
class Node:
    ..def init(self, val):
        ....self.val = val
```

```
....self.next = None
```

Lista zawierała liczby naturalne stanowiące kolejne wyrazy rosnącego ciągu arytmetycznego. Pewien fragment listy zastąpiono jego rewersem. Na szczęście pierwszy i ostatni element ciągu pozostały na swoich miejscach. Proszę napisać funkcję `repair(p)`, (`p` wskazuje na pierwszy element listy) która przywraca listę do stanu początkowego. Funkcja powinna zwrócić długość odwróconego fragmentu.

Na przykład: listę 3,11,9,7,5,13,15,17,19 należy przekształcić w listę 3,5,7,9,11,13,15,17,19

**Zadanie 234.** Dana jest definicja klasy, której obiekty stanowią elementy listy odsyłaczowej:

```
class Node:
..def init(self,val,next=None):
....self.val = val
....self.next = next
```

Lista zawiera wartości będące liczbami naturalnymi. Proszę napisać funkcję `divide(p)`, (`p` wskazuje na pierwszy element listy) która rozdziela listę na dwie listy. W pierwszej powinny się znaleźć liczby, które są wielokrotnością (co najmniej dwukrotnością) kwadratu dowolnego wyrazu ciągu Fibonacciego większego od 1, a w drugiej pozostałe liczby. Względny porządek w powstałych listach nie powinien ulec zmianie. Funkcja powinna zwrócić wskazania do obu list.

## Zestaw 7: Struktury drzewiaste

**Zadanie 235.** Proszę napisać następujące funkcje:

1. Funkcja wypisującą zawartość drzewa.
2. Funkcja, która sprawdza czy dana liczba należy do drzewa.
3. Funkcja, która zwróci rozmiar drzewa (liczbę węzłów).
4. Funkcja, która zwróci wysokość drzewa (liczbę poziomów).
5. Funkcja, która zwróci liczbę liści drzewie.
6. Funkcja, która zwróci liczbę węzłów na n-tym poziomie.
7. Funkcja, która zwróci liczbę węzłów mających jednego potomka.

Każdą funkcję proszę zrealizować jako funkcję rekurencyjną, a następnie jako funkcję bez użycia rekurencji.

**Zadanie 236.** Proszę napisać następujące funkcje:

1. Funkcja, która usunie wszystkie węzły drzewa.
2. Funkcja, która usunie węzły powyżej n-tego poziomu.
3. Funkcja, która usunie wszystkie aktualne liście drzewa.

**Zadanie 237.** Proszę napisać funkcję, która sprawdza czy dane drzewo jest poprawnym drzewem BST.

**Zadanie 238.** Proszę napisać funkcję, która sprawdza czy dana liczba należy do drzewa BST. Funkcję proszę zrealizować jako funkcję rekurencyjną, a następnie jako funkcję bez użycia rekurencji.

**Zadanie 239.** Proszę napisać funkcję, która wstawi liczbę do drzewa BST. Funkcję proszę zrealizować jako funkcję rekurencyjną, a następnie jako funkcję bez użycia rekurencji.

**Zadanie 240.** Dane jest drzewo BST zbudowane z elementów ... Proszę zaimplementować funkcję zwracającą długość najdłuższej ścieżki kończącej się liściem, w której każdy węzeł ma tylko jednego syna.

**Zadanie 241.** Proszę napisać funkcję, która dla drzewa BST zwraca wysokość najwyższego poddrzewa będącego listą.

**Zadanie 242.** W kodzie Morse'a litery kodowane są za pomocą popularnie zwanych kropką i kreską, np. 'A': '-.', 'B': '-...', 'C': '-.-.', itd. Proszę napisać funkcję, która tworzy drzewo binarne (na lewo kropka na prawo kreska) pozwalające dekodować znak poprzez schodzenie w drzewie od korzenia do odpowiedniego węzła. Następnie proszę napisać funkcję dekodującą napis złożony ze znaków kropki i kresek.

**Zadanie 243.** Drzewo Hufmanna.

## Zestaw 8: Wyszukiwanie i sortowanie

**Zadanie 244.** Dana jest duża tablica zawierająca liczby naturalne posortowane niemalejąco. Proszę napisać funkcję, która zwraca liczbę wystąpień danego elementu w tablicy.

**Zadanie 245.** Proszę napisać funkcję, wyszukującą element w posortowanej rosnąco tablicy, w której zamiast podziału na dwa zastosowano podział na trzy części. Proszę ocenić szybkość działania wyszukiwania w porównaniu z klasycznym wyszukiwaniem połówkowym.

**Zadanie 246.** Dana jest tablica  $T[N]$  wypełniona liczbami nieujemnymi. Wartości 0 oznaczają dziury w których nie można przechowywać liczb naturalnych. Proszę napisać funkcję sortującą taką tablicę.

**Zadanie 247.** Dany jest ciąg kwadratowych klocków. Każdy klocek ma kształt kwadratu pokolorowanego 4 kolorami. Klocki w ciągu mogą być dowolnie ustawione (przekręcone), a nawet odwrócone. Proszę napisać funkcję porządkującą klocki, tak aby identyczne klocki występowały obok siebie. Ciąg klocków jest reprezentowany jako tablica krotek, np.  $T = [(2,3,1,4), (3,2,4,1), (1,2,3,4), \dots]$

**Zadanie 248.** Dana jest lista odsyłaczowa przechowująca liczby całkowite. Proszę napisać funkcję wstawiającą istniejący element do posortowanej listy, a następnie zrealizować funkcję sortowania przez wstawianie (insertion sort).

**Zadanie 249.** Dana jest lista odsyłaczowa przechowująca liczby całkowite. Proszę napisać funkcję wypinającą maksymalny element z listy, a następnie zrealizować funkcję sortowania przez wybieranie (selection sort).