Zadania (vector i string)

Zadanie 1. Napisz program, który wczytuje kolejne liczby całkowite, aż do napotkania znaku końca pliku (Ctr-Z), a następnie drukuje ich sumę.

Zadanie 2. Napisz program, który wczytuje kolejne liczby całkowite, aż do napotkania znaku końca pliku (Ctr-Z), a następnie drukuje ich medianę.

Zadanie 3. Napisz program, który wczytuje najpierw liczbę naturalną n, następnie wczytuje n liczb typu double, a potem wyświetla je w odwrotnej kolejności.

Zadanie 4. Napisz funkcję, która sortuje wektor elementów typu int.

Zadanie 5. Napisz funkcję, która oblicza sumę wektora elementów typu double. Zakładamy, że na wejściu funkcji wektor jest niepusty. Przekaż argument w optymalny sposób (tzn. przez referencję na stałą).

Zadanie 6. Napisz funkcję, która oblicza medianę wektora elementów typu int. Zakładamy, że na wejściu funkcji wektor jest niepusty. Funkcja ma nie modyfikować wejściowego wektora.

Zadanie 7. Napisz funkcję substr o trzech argumentach: s typu string oraz n i m typu int. Wartością funkcji ma być podłańcuch s[n]..s[m]. Możesz założyć, że $0 \leq n \leq m \leq$s.size()-1.
Przykład: substr("abcde", 1, 3) -> "bcd"

Zadanie 8. Napisz funkcję, której argumentem jest element typu string, a wartość jest typu bool. Funkcja ma sprawdzać, czy dany łańcuch jest palindromem, tzn. jeśli jest, to ma zwracać wartość true, w przeciwnym razie false.
Przykład: kajak to palindrom, a rabarbar nie.

Zadanie 9. Ciąg słów Fibonacciego definiujemy następująco:

$$\begin{aligned} Fib_0 &= b, \\ Fib_1 &= a, \\ Fib_n &= Fib_{n-1}Fib_{n-2} \text{ dla } n \geq 2. \end{aligned}$$

Kilka kolejnych słów Fibonacciego to:

$$b, a, ab, aba, abaab, abaababa, abaababaabaab, \ldots$$

Napisz funkcję, która dla danego n oblicza i zwraca n-te słowo Fibonacciego. Funkcja może, ale nie musi, być rekurencyjna.

Zadanie 10. Napisz program, który wczytuje kolejne łańcuchy tekstowe (rozdzielone białymi znakami, aż do napotkania znaku końca pliku), sortuje je wg. rosnącej długości i wyświetla.

Zadanie 11. Napisz program, który wczytuje kolejne *linie* tekstu (rozdzielone znakami końca linii, aż do napotkania znaku końca pliku), sortuje je leksykograficznie i wyświetla.
Wskazówka: Aby wczytać wszystkie znaki od danej pozycji do znaku nowej linii (łącznie ze spacjami), należy zamiast cin>>s użyć funkcji getline(cin, s), gdzie s jest zmienną typu string.
Uwagi:
1. Funkcja getline(cin, s) nie pomija wiodących białych znaków.
2. Funkcja getline(cin, s) pobiera znak nowej linii ze strumienia, ale go nie dodaje do s.

Zadanie 12. Zdefiniuj element v typu vector<vector<**int**> > i zainicjalizuj go tak, aby jedynymi jego elementami były

$$v[0][0] == 1, \quad v[0][1] == 0, \quad v[0][2] == 0,$$
$$v[1][0] == 0, \quad v[1][1] == 2, \quad v[1][2] == 0.$$

Spróbuj wykonać powyższe zadanie na różne sposoby.

Zadanie 13. Zdefiniuj funkcję, która przyjmuje trzy argumenty: v typu vector<vector<**int**> > oraz m i n typu int. Efektem działania funkcji, ma być zainicjalizowanie elementu v tak, aby reprezentował on macierz zerową o wymiarach m na n (precyzyjnie, v ma być wektorem o rozmiarze m, a każdy jego element ma być wektorem o rozmiarze n, elementów typu int, zainicjalizowanym zerami).

Zadanie 14. Napisz program, który dla zadanego n, generuje trójkąt Pascala, a następnie go wyświetla. Kolejne wiersze mają być reprezentowane przez element typu vector<**int**>, a cały trójkąt przez vector<vector<**int**> >.