**Приклади екзаменаційних задач з функціонального програмування. 2022**

1. Створити чергу з алфавітно-цифрових символів. Побудувати стек з тих елементів черги, які є голосними латинськими літерами. Цифрові символи з черги записати в окремий список. Вивести на екран побудовані стек, список та кількість елементів в них
2. Розв’язати нелінійне рівняння *x* = cos(*x*) будь-якими чисельними методами (бісекції, перебору, хорд, дотичних) визначивши інтервал [*a, b*], на якому існує рішення рівняння. Значення *a, b* інтервалу вибрати самостійно. Порівняти результати розв’язків двома методами.
3. Створити список натуральних чисел (натуральні числа >= 1), кратних 3, задавши їх кількість. Вивести створений список. Додати елементи в список на задану позицію в списку; підрахувати кількість парних елементів в списку.
4. Задати текст. Текст представляти списком списків: кожне речення є список слів, весь текст є список речень. Необхідно в заданому тексті видалити з кожного слова в кожному реченні всі літери, що повторюються. З видалених літер утворити новий список.
5. Створити список парних натуральних чисел (натуральні числа >=1), задавши їх кількість. Вивести створений список. Додати задану кількість елементів на початок списку. Здійснити пошук заданого елемента та визначення позиції його в списку, у випадку відсутності елемента вивести відповідне повідомлення.
6. Створити список непарних натуральних чисел (натуральні числа >=1), задавши їх кількість. Вивести створений список. Інвертувати вхідний список із заданої позиції до кінця (до заданої позиції) списку; сформувати новий список із елементів, які є квадратами чисел вхідного списку.
7. Створити список факторіалів цілих чисел, задавши їх кількість. Вивести створений список. Визначити максимальний та мінімальний елементи списку; вставити в список задану кількість елементів між знайденими максимальним і мінімальним елементами
8. Створити вектор. Обчислити суму елементів між максимальним та мінімальними значеннями вектору. Вивести на екран максимальний, мінімальний елементи вектору, їх індекси та шукану суму елементів
9. Створити вектор чисел з додатними, від’ємними та нульовими елементами. Переставити елементи вектору так, щоб спочатку були розташовані всі від’ємні елементи, потім усі додатні елементи, потім усі нульові. Вивести вектор на екран
10. Створити список випадкових цілих чисел, задавши їх кількість. Вивести створений список. Визначити елементи списку, які є простими числами. Замінити кожне третє число в списку на значення, введене з клавіатури
11. Створити два списки цілих чисел. Вивести їх на екран. Побудувати третій список, в якому кожний елемент дорівнює найбільшому спільному дільнику відповідних елементів введених списків. Визначити елементи, які співпадають в обох списках та надрукувати їх.
12. Знайти корені нелінійного рівняння виду ***x*2 = e-*x***. Пошук наближеного значення хоча б одного кореня рівняння ***f*(*x*) = 0** на відрізку [*a*; *b*] здійснювати методами перебору та хорд. Значення *a, b* інтервалу задати з клавіатури. Обчислити похибку результатів розв’язків двома методами.
13. Задати текст, який поданий списком списків: кожне речення є список слів, весь текст є список речень. Написати програму, яка створює список, в якому в кожному слові кожного речення замість символів, що повторюються, вказується кількість повторень символів в слові. Контрольний тест: '((aaabb ccccddd)(eeefgg hhkl)) перетворюється в '(а3b2 c4d3)(e3fg3 h2kl))
14. Побудувати стек натуральних чисел. Вивести на екран створений стек. Надрукувати в зворотному порядку числа стеку, пропускаючи кратні заданому з клавіатури користувачем числу. З пропущених чисел утворити список
15. Дек – це такий послідовний список, в якому як включення, так і видалення елементів може здійснюватися з будь-якого з двох його кінців. Створити дек з чисел *xi*, i = 1,…,n. Визначити кількість та надрукувати значення елементів дека. Якщо дек має парну кількість елементів, обчислити добуток різниць елементів , інакше суму різниць елементів 
16. Побудувати два однозв’язних списків чисел. Порівняти списки. Якщо списки не однакові, виконати їх конкатенацію, інакше утворити два нових списки, значення елементів яких є парними та непарними числами
17. Увести рядок алфавітних, цифрових символів, довільну кількість символів '(' та ')'. Дужки можуть бути вкладеними. Якщо дужки складають правильний дужковий вираз, то вивести на екран усі символи, розташовані всередині пари дужок, і підрахувати їх кількість. Записати в новий список символи, розташовані всередині пари дужок. Якщо дужки складають неправильний дужковий вираз (їх кількість непарна, дужки записані в неправильній послідовності), вивести відповідне повідомлення.
18. Написати програму, яка по заданому тексту будує список пар: (<слово> <частота повторення в тексті>). Текст представляти списком списків: кожне речення є список слів, весь текст є список речень, речення завершується символом «крапка». З трьох слів, що повторюються найчастіше, утворити новий список.
19. На вокзалі працює **k** кас, проте черга до них одна. Коли усі каси вільні, перші **k** клієнтів з черги підходять до кас. Інші чекають своєї черги. Як тільки кого-небудь буде обслужено і відповідна каса звільниться, наступна людина з черги підходить до цієї каси. Так продовжується до тих пір, доки не буде обслужено усіх клієнтів. Задати час обслуговування одного клієнта з клавіатури. Визначити час, за який буде обслужено усіх клієнтів
20. Заданий вектор як послідовність чисел, і матриця як послідовності векторів (рядів матриці). Напишіть процедури для обчислення матричних операцій складання, віднімання матриць, множення матриці на вектор. Передбачити випадки невідповідності вимірності матриць і векторів.
21. Побудувати програму «стиснення» для перетворення англійських слів відповідно до наступних правил: перша буква слова зберігається; всі наступні за нею голосні, видаляються; здвоєні букви замінюються одиночними. Стиснене слово складається не більше ніж з чотирьох букв, інші літери видаляються.
22. Створити числовий список, кількість елементів якого задана користувачем. Значення елементів в списку можуть повторюватися. Вивести список на екран. Визначити позицію першого та останнього входження заданого числа в список. Знайти елемент, значення якого повторюється найбільшу кількість разів.
23. Граф з кінцевою кількістю вершин представлений парою з кількості вершин і функції, яка видає True, якщо аргументи представляють собою номери вершин, з'єднаних ребром. Скласти програму, яка перевіряє, чи є в графі маршрут, що єднає дві задані вершини, застосувавши алгоритм обходу графа в ширину. Надрукувати граф списком ребер та результат обходу графа.
24. Структура типу дерева являє собою кореневий вузол, що містить деяке значення довільного типу, і список піддерев. Написати функцію, яка обчислює висоту дерева.
25. *N* десантників-миротворців приземлилися. Точка приземлення кожного десантника задається цілочисельними координатами (*х, у*). За один крок кожен з десантників може переміститися на сусідню цілочисельну позицію уздовж осі *Х* або *Y* (тобто одна з його координат міняється на 1 за абсолютною величиною). Кроки робляться по черзі, ніякі два миротворці при цьому не можуть знаходитися в одній позиції одночасно. *N* десантників хочуть вишикуватися в шеренгу – лінію, паралельну одній з осей координат, в якій вони стояли б в підряд розташованих цілочисельних позиціях. Напишіть програму, яка визначає мінімальну сумарну кількість кроків, необхідних десантникам для того, щоб утворити шеренгу.
26. Структура графа задана списками суміжності номерів вершин, тобто списком, елементами якого є пари, що складаються з номера вершини і списку номерів вершин, інцидентних їй. Написати функцію, яка перевіряє, чи існує в графі шлях, що з’єднує вершини із заданими номерами.