**Екзаменаційні задачі з функціонального програмування**

**(2020)**

* 1. Написати процедуру обчислення кореня квадратного методом наближення Ньютона, за яким . Основна процедура визначена від двох аргументів: наближення і числа, квадратний корінь якого вона знаходить. Якщо поточне наближення є досить хорошим, то процес обчислення закінчується і результатом стає поточне наближення, інакше обчислення триває з поліпшеним значенням наближення. Для визначення кореня квадратного від числа визначати допустиму похибку обчислення як модуль різниці наближених значень кореня на попередньому та поточному кроках у вигляді предиката. Не використовувати цикли.
  2. Написати процедуру, яка для заданого списку lst формує список-результат шляхом об'єднання результату інвертування lst, результату інвертування хвоста lst, результату інвертування хвоста lst і так далі. Приклад: для списку (1 2 3 4 5 6) результатом буде:(6 5 4 3 2 1 6 5 4 3 2 6 5 4 3 6 5 4 6 5 6). В отриманому списку визначити усі парні числа, вивести їх в окремий список і визначити їх кількість.
  3. 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. З трьох слів, що повторюються найчастіше, утворити новий список.
  4. Написати програму, яка в заданому тексті видаляє з кожного слова в кожному реченні всі літери, що повторюються. З видалених літер утворити новий список. Текст представляти списком списків: кожне речення є список слів, весь текст є список речень.
  5. Написати програму, що сортує список цілих чисел за алгоритмом Хоара. Алгоритм Хоара розбиває список на дві частини щодо опорного елемента, потім рекурсивно сортує кожну частину списку. Масиви не використовувати.
  6. Написати процедуру мовою Scheme з двома параметрами: атомом і списком цілих чисел. Якщо список пустий, то вивести відповідне повідомлення. Якщо список не пустий, процедура повертає список, з якого вилучено заданий атом, незалежно від глибини його знаходження в списку. Список, що повертається, не має на місці вилученого атома ніякого елемента. В отриманому після вилучення атома списку поміняти місцями парні і непарні числа.
  7. Написати процедуру мовою Scheme, яка отримує список в якості параметра і повертає його, видаливши з нього другий від початку та третій з кінця елементи. Якщо заданий список пустий або містить менше двох елементів, процедура повинна повертати порожній список ().
  8. Задати текст, який поданий списком списків: кожне речення є список слів, весь текст є список речень. Написати програму, яка в кожному слові кожного речення для повторюваних літер робить таку заміну: повторні входження літер видаляє, до першого входження літери приписує кількість входжень літери в слово. Наприклад: '((aaabb ccccddd)(eeefgg hhkl)) перетворюється в '(а3b2 c4d3)(e3fg3 h2kl))
  9. Число x називається нерухомою точкою (fixed point) функції f, якщо воно задовольняє рівнянню f (x) = x. Написати процедуру, яка в якості аргументів приймає функцію і початкове значення і виробляє наближення до нерухомої точки функції. Для деякої функції f можна знайти нерухому точку, почавши з якогось значення і застосовуючи f багаторазово, поки значення не перестане змінюватися відповідно до заданої точності.
  10. Множину можна уявити як список його різних елементів, а множину його підмножин як список списків. Використовуючи форми let і lambda, написати процедури, перетину, об’єднання та віднімання множин.
  11. Написати програму, яка виключає в початковому тексті з кожного слова його закінчення за словником. Словник закінчень представляти списком рядків. Текст представляти списком списків: кожне речення є список слів, весь текст є список речень.
  12. Написати програму, яка створює бінарне дерево та обчислює суму його листків, кількість рівнів та найбільше значення в його вершинах.
  13. Сто чоловік стоять у черзі так, що кожен бачить тільки тих, хто стоять перед ним, але чує усе, що говорять інші. На кожну людину одягають ковпак одного з трьох кольорів (скажімо, червоного, синього або білого), колір, зрозуміло, йому невідомий. Починаючи з останнього в черзі (того, хто бачить усіх), у кожної людини запитують, якого кольору у неї ковпак. Якщо вгадує - його залишають в черзі, помиляється - його вилучають. Необхідно придумати, як домовиться цим 100 людям, щоб мінімізувати жертви.
  14.  Дек - це такий послідовний список, в якому як включення, так і виключення 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 суму його елементів. Інакше визначити значення виразу
  15. **.** Йде посадка в літак, що розрахований на 20 місць. В черзі на посадку стоять 20 пасажирів. Першим входить старий чоловік. Зайшовши в салон, він сідає на будь-яке випадково вибране місце. Кожний з наступних пасажирів, зайшовши в салон, сідає на своє (позначене в білеті) місце, якщо воно вільне, і на будь-яке з вільних - в протилежному випадку. Яка ймовірність, що останній в черзі пасажир сяде на своє місце?
  16. Побудувати два однозв’язних списків. Порівняти списки. Якщо списки не однакові, виконати їх конкатенацію, інакше утворити новий список, значення елементів яких є парними або непарними числами
  17. Напишіть процедуру, яка повертає число різних пар в списку. (Підказка: переглядайте структуру, підтримуючи при цьому допоміжну структуру, що стежити за тим, які пари вже були пораховані.) Здійснити видалення елемента за його значенням та за його порядковим номером у списку. У разі відсутності елемента із заданим значенням видати відповідне повідомлення.
  18. Написати процедуру , що для аргументу-списку формує список-результат за правилом: якщо перший й останній елементи списку-аргументу є парні числа, то включити в список-результат квадрати чисел списку-аргументу, інакше сформувати список з першого й останнього елементів.
  19. Написати програму мовою Scheme для визначення коренів нелінійного рівняння виду **x2=e-x**. Пошук наближеного значення хоча б одного кореня рівняння ***f*(*x*) = 0** на відрізку [*a*; *b*] здійснювати методами перебору та хорд
  20. Написати програму, яка створює бінарне дерево, що містить в якості елементів символи. Необхідно написати процедуру, яка по дереву повертає рядок, що зображує це дерево, а також процедуру, яка бере в якості параметра рядок, і повертає дерево. Для простоти можна вважати, що в дереві не зустрічаються символи '(' і ')'. І можна вважати, що рядок правильний, він дійсно відповідає деякому дереву.
  21. Написати програму, яка створює бінарне дерево, перераховує листя дерева; просіює їх, відбираючи непарні; зводить в квадрат кожне з відібраних чисел і знаходить їх суму.
  22. Написати програму, яка створює послідовність чисел в довільному порядку, сортує їх методом бульбашкового сортування, знаходить в послідовності числа Фібоначчі, просіює їх, вибираючи непарні і конструює з них список. Якщо чисел Фібоначчі в послідовності нема, вивести відповідне повідомлення.
  23. Припустимо, що вектори v = (vi) є послідовності чисел, а матриці m = (mij) представлені як послідовності векторів (рядів матриці). Напишіть процедури для обчислення матричних операцій складання і віднімання матриць, множення матриць на вектор. Передбачити випадки невідповідності вимірності матриць і векторів.
  24. Написати процедуру , що для аргументу-списку формує список-результат за правилом: якщо перший й останній елементи списку-аргументу є парні числа, то включити в список-результат квадрати чисел списку-аргументу, інакше сформувати список з першого й останнього елементів.
  25. Створити телефонний довідник у вигляді черги, елементами якої є дані про абонента телефонної компанії (прізвище, номер телефону). Здійснити пошук абонента за його прізвищем та за номером. У разі відсутності даних вивести відповідне повідомлення. Надрукувати телефонний довідник у вигляді таблиці
  26. Процедура accumulate відома також як fold-right (права згортка), оскільки вона комбінує перший елемент послідовності з результатом комбінування всіх елементів праворуч від нього. Існує також процедура fold-left (ліва згортка), яка подібна до fold-right, але комбінує елементи в протилежному напрямку: