

Задание 1

- Представление массива в виде списка.** Описать список – “Разреженный Массив”. Каждый элемент списка хранит индекс и значение ненулевого элемента массива. Нулевые элементы массива в списке не хранятся. Первый элемент списка хранит размер массива. Описать функции
 - Создать массив. Элементы массива подаются из текстового файла. Данные в файле заданы следующим образом. В первой строке задано целое число n – размер массива. В следующей строке заданы n вещественных чисел – элементы массива (и нулевые и ненулевые);
 - извлечение элемента массива по индексу;
 - запись в массив значения по индексу;
 - сложение двух массивов;
 - скалярное произведение двух массивов;
 - вывод массива в текстовый файл (массив выводится полностью, с нулевыми элементами) в формате “имя массива, знак равенства, значение массива”.
- С использованием функций решить следующую задачу. Ввести два массива a, b из текстового файла. Найти массив $c = (a + b)(a * b)$. Результирующий массив c вывести на экран.

Задание 3

- Представление массива в виде списка.** Функциональная сумма вида $P(x) = \sum_{i=0}^r p_i e^{-i*x}$ задается в виде списка коэффициентов $\{p_i\}$. В списке хранятся только ненулевые коэффициенты. Описать функции:
 - Создать список из текстового файла. Данные в файле заданы следующим образом. В первой строке задано целое число r . В следующей строке заданы r вещественных чисел – коэффициенты $\{p_i\}$ функциональной суммы (и нулевые и ненулевые);
 - Вывод функциональной суммы в текстовый файл (в отформатированном виде);
 - Умножение двух функциональных сумм;
 - Сложение двух функциональных сумм;
 - Вычисление значения суммы в точке $x = a$;
- С использованием описанных функций составить программу для решения следующей задачи:
 - Создать две функциональные суммы $P_1(x), P_2(x)$. Значения взять из текстовых файлов;
 - Построить и вывести в текстовый файл функциональную сумму $P_4(x) = (P_1(x) + P_2(x)) * P_1(x)$. Вывести результат в текстовый файл;

- (с) Вычислить сумму $P_4(x)$ в точке $x = a$ (значение a вводится с клавиатуры). Вывести результат на экран.

Возможно использования других функций, если в этом возникает необходимость.

Задание 4

1. **Представление матрицы в виде списка.** Описать список `Matr` – “Разреженная стохастическая матрица”. Матрица называется стохастической, если каждый ее элемент находится в диапазоне $[0, 1]$ и сумма элементов каждой строки равна 1. Матрица представляется в виде списка следующим образом. Каждый элемент списка хранит индексы (номер строки и номер столбца) и значение ненулевого элемента матрицы. Нулевые элементы матрицы в списке не хранятся. Первый элемент списка хранит размерность матрицы (матрица квадратная). Описать функции:
 - (а) Создание стохастической матрицы в виде списка. Данные подаются из текстового файла. В файле записаны: в первой строке - размер матрицы. Затем n строк по n вещественных чисел, задающих элементы каждой строки матрицы (нулевые элементы в файле присутствуют);
 - (б) проверить матрицу на дважды стохастичность. (Матрица называется дважды стохастической, если каждый ее элемент находится в диапазоне $[0, 1]$ и сумма элементов каждой строки и каждого столбца равна 1);
 - (с) найти максимальный элемент j -го столбца матрицы;
 - (d) найти номер столбца, содержащего только нулевые элементы;
 - (е) удалить из матрицы i -ую строку и i -ый столбец с пересчетом всех индексов;
 - (f) вывести матрицу в текстовый файл. Матрицу выводить в отформатированном виде с разбиением на строки. Нулевые элементы тоже выводятся.
2. С использованием данных функций решить следующую задачу.
 - (а) Ввести стохастическую матрицу из текстового файла.
 - (б) Удалить все нулевые столбцы и соответствующие им строки.
 - (с) Вывести результирующую матрицу в текстовый файл.
 - (d) Проверить ее на дважды стохастичность.
 - (е) Вывести номер столбца, содержащего максимальный элемент.

Возможно наличие других функций, если в этом возникает необходимость.

Задание 7

1. **Представление матрицы в виде списка.** Описать список `Matr` – “Разреженная целочисленная матрица”. Каждый элемент списка хранит индексы (номер строки и номер столбца) и значение ненулевого элемента матрицы. Нулевые элементы матрицы в списке не хранятся. Первый элемент списка хранит размерность матрицы (число строк и число столбцов). Описать функции:

- (a) Создание матрицы. Данные подаются из текстового файла. В файле записаны: в первой строке – размер матрицы (n – число строк и m – число столбцов). Затем n строк по m целых чисел (нулевые элементы в файле присутствуют);
- (b) найти минимальный элемент i -ой строки матрицы;
- (c) найти максимальный элемент j -го столбца матрицы;
- (d) вывод матрицы в текстовый файл (с разбивкой на строки). Нули также выводятся;

2. С использованием данных функций решить следующую задачу.

- (a) Создать “Разреженную целочисленную матрицу” A из текстового файла;
- (b) Для заданной матрицы A вывести на экран номера и элементы тех строк, в которых минимальный элемент является максимальным в своем столбце.
- (c) Затем в каждой строке каждый элемент, находящийся после минимума (если минимум не является последним элементом в столбце) заменить на номер столбца этого элемента.
- (d) Вывести полученную матрицу в текстовый файл.

Возможно наличие других функций, если в этом возникает необходимость.

Задание 8

1. **Списки слов.** Описать список Words – “Список слов, начинающихся с одной и той же буквы”. Каждый элемент списка хранит слово (динамический массив) и целое число, соответствующее количеству вхождений данного слова в текст. Описать функции:
 - (a) Создать список (создается пустой список);
 - (b) Положить слово в список (если такого слова в списке не было, добавляется новый элемент с данным словом, количество – 1; если слово в списке присутствовало, оно не добавляется, количество соответствующего слова увеличивается на единицу);
 - (c) найти слово с максимальной длиной;
 - (d) найти максимальное и минимальное количество вхождений какого-либо слова в текст;
 - (e) Отсортировать список по уменьшению количества вхождений слов в текст;
 - (f) Удалить список.
2. С использованием данных функций решить следующую задачу.
 - (a) Создать массив списков для хранения слов текста. Каждый список в массиве хранит слова, начинающиеся с одной и той же буквы. Соответственно, размер массива равен числу букв алфавита (возьмем для определенности английский алфавит и соответственно, англоязычные слова);

- (b) Для заданного в файле текста составить частотный словарь слов. Все слова текста помещаем в соответствующие списки, где попутно считаем количество их присутствия в тексте.
- (c) С использованием построенного списка вывести все слова такие, которые наиболее часто и наименее часто встречаются в тексте.
- (d) Вывести самое длинное слово и количество его присутствия в тексте.
- (e) Вывести все слова, начинающиеся на заданную букву в порядке убывания количества вхождений слов в текст.
- (f) Удалить все списки.

Возможно наличие других функций, если в этом возникает необходимость.

Задание 9

1. **Список символов.** Описать список `Symb` – “Список символов текста”. Каждый элемент списка хранит символ и целое число, соответствующее количеству вхождений данного символа в текст. Описать функции:
 - (a) Создать список. (создается пустой список).
 - (b) Положить символ в список (если такого символа в списке не было, добавляется новый элемент с данным символом, количество – 1; если символ уже в списке присутствует, он не добавляется, количество соответствующего символа увеличивается на единицу);
 - (c) найти символ с максимальным числом вхождений в текст;
 - (d) найти максимальное и минимальное количество вхождений какого-либо символа в текст и их частоту вхождения (количество вхождений / количество слов);
 - (e) Отсортировать список по уменьшению количества вхождений символов в текст;
 - (f) Удалить список.
2. С использованием данных функций решить следующую задачу.
 - (a) Создать список для хранения символов текста;
 - (b) Для заданного в файле текста составить частотный словарь символов. Все символы текста помещаем в список, где попутно считаем количество их присутствия в тексте;
 - (c) Вывести все наиболее часто встречающиеся символы;
 - (d) С использованием построенного списка вывести все символы в порядке убывания частоты их присутствия в тексте. Выводить символ и частоту его вхождения в текст;
 - (e) Вывести список в текстовый файл;
 - (f) Удалить список;
 - (g) Протестировать программу на различных текстах и сравнить полученные результаты по частотному анализу символов.

Возможно наличие других функций, если в этом возникает необходимость.

Задание 10

1. **Список пар символов.** Описать список `ParSymb` – “Список пар символов текста”. Каждый элемент списка хранит 2 символа и целое число, соответствующее количеству вхождений данной пары символов в текст (присутствие пары символов рядом в тексте). Описать функции:
 - (a) Создать список. (создается пустой список).
 - (b) Положить пару символов в список (если такой пары в списке не было, добавляется новый элемент с данной парой, количество – 1; если такая пара символов уже присутствует в списке, она не добавляется, количество соответствующей пары увеличивается на единицу);
 - (c) найти наиболее часто встречающиеся в тексте пары символов;
 - (d) найти максимальное и минимальное количество вхождений какой-либо пары в текст и их частоту вхождения (количество вхождений / количество пар в тексте);
 - (e) Отсортировать список по уменьшению количества вхождений пар символов в текст;
 - (f) Вывести список в текстовый файл.
 - (g) Удалить список.
2. С использованием данных функций решить следующую задачу.
 - (a) Создать список для хранения пар символов текста;
 - (b) Для заданного в файле текста составить частотный словарь пар символов. Все пары символов текста помещаем в список, где попутно считаем количество их присутствия в тексте;
 - (c) Вывести все наиболее часто встречающиеся пары символы;
 - (d) С использованием построенного списка вывести все пары символы в порядке возрастания частоты их присутствия в тексте. Выводить символ и частоту его вхождения в текст;
 - (e) Удалить список;
 - (f) Протестировать программу на различных текстах и сравнить полученные результаты по частотному анализу символов.

Возможно наличие других функций, если в этом возникает необходимость.

Задание 11

Тригонометрический многочлен $a_0 + a_1 \cos x + b_1 \sin x + a_2 \cos 2x + b_2 \sin 2x + \dots + a_n \cos nx + b_n \sin nx$ задается последовательностью коэффициентов;

Программа должна содержать следующие функции:

1. кодирования: создание списка, элементы которого содержащего тройки (n , коэффициент при $\cos nx$, коэффициент при $\sin nx$) по вектору, заданному в текстовом файле;

2. декодирования: восстановления вектора коэффициентов с выводом результата в текстовый файл, с освобождением выделенной динамической памяти;
3. вставки элемента в список: вставки ненулевого элемента в некоторую позицию (позиция определяется в интерактивном режиме); при этом если таковая компонента в векторе существовала, то происходит замена этой компоненты;
4. удаления элемента из списка;
5. преобразование: поиск производной;
6. преобразование: нахождения суммы двух многочленов;
7. вычисление: вычисления значения многочлена в некоторой точке;
8. преобразование: построения нового многочлена, у которого коэффициенты при \cos так и при \sin одновременно не равны нулю.

Написать программу, демонстрирующую работу данных функций.

Задание 12

Полином от трех переменных представить в виде списка коэффициентов полинома. Каждый элемент содержит значение коэффициента и значение показателей степеней переменных при этом коэффициенте. Вектора показателей степеней упорядочены в лексикографическом порядке следования переменных. Программа должна содержать следующие функции:

1. кодирования: построение списка по полиному, заданному в текстовом файле. Обдумать и обосновать внешний формат исходных данных;
2. декодирования: восстановления исходного вида полинома с отформатированным выводом результата в текстовый файл;
3. уничтожения списка с освобождением выделенной динамической памяти;
4. вставки элемента в список: вставки элемента в некоторую позицию списка (при вставке учесть существует ли подобный элемент в списке, если да – то заменить существующий);
5. удаления элемента из списка: для заданного вектора показателей степеней удаление из списка элемента с четными коэффициентами;
6. Проверки для вектора показателей степеней, существует ли элемент списка с данными значениями показателей;
7. Вычислить значение полинома в некоторой точке;
8. Найти вектор степеней слагаемого многочлена, имеющий минимальный коэффициент

Написать программу, демонстрирующую работу данных функций.

Задание 13

Хеш-таблица *Hashable* представляет из себя массив фиксированного размера m (по определению m меньше мощности множества U значений, которые мы в эту таблицу можем сохранить, целью использования хеш-таблицы для хранения данных является минимизация памяти при эффективности выполнения основных операций). При добавлении элемента x в таблицу *Hashable* значение x сохраняется в ячейке по номером $h(x)$, где $h : U \rightarrow \{0, \dots, m - 1\}$ – хеш-функция. Для эффективной работы с хеш-таблицей хеш-функция должна удовлетворять определенным свойствам, среди которых свойство равномерного распределения данных по хеш-таблице, с целью минимизации вероятности ситуации когда для двух различных значений x, x' значения $h(x)$ и $h(x')$ совпадают. Эта ситуация называется коллизией. Поскольку размер таблицы меньше числа потенциальных значений, которые мы в данную таблицу можем сохранить, в общем случае коллизии избежать не удастся, и следовательно, эту проблему придется как то решать. Одно из возможных решений проблемы коллизии следующее: при добавлении нескольких элементов в одну и ту же ячейку таблицы, их связывают в список. Таким образом – хеш-таблица, это таблица размера m , каждая ячейка которой хранит список, хранящий элементы, у которых хеш-значение совпадает.

Описать функции:

1. создать хеш-таблицу;
2. добавить элемент в хеш-таблицу;
3. удалить элемент из хеш-таблицы;
4. поиск элемента в хеш-таблице;
5. объединить две хеш-таблицы;
6. удалить хеш-таблицу;
7. проверить две таблицы на совпадение;
8. скопировать хеш-таблицу;
9. перестроить хеш-таблицу в новую с изменением ее размера и хеш-функции.

С использованием описанных функций составить программу для решения следующей задачи. Даны два текстовых файла “in1.txt, in2.txt”. В первом файле хранятся ключевые слова, во втором файле – произвольный текст. Создать таблицу $T1$, в которую поместить все слова-ключи. Вывести все слова файла “in2.txt”, совпадающие с ключами. Создать две новые таблицы $T2, T3$ из данных файлов “in3.txt, in4.txt”. Проверить совпадение таблиц $T1, T2$. Если они не совпадают, объединить их в таблицу $T4$. Если данные в $T4$ и в $T3$ не совпадают, то удалить из $T3$ элементы $T4$. Параметры создаваемых таблиц вводятся с клавиатуры. После каждого шага содержимое таблиц выводить на экран. Хеш-функция: сложение кодов символов строки по модулю m .