Темы курсовой работы

Студент может выбрать тему курсовой работы из приведенного ниже списка или предложить свою тему, согласовав ее с преподавателем.

*1. «Словарь с дубликатами»*

Словарь с дубликатами – это структура данных, обеспечивающая доступ к информации по ключу и допускающая хранение элементов данных с одинаковыми ключами. Элемент словаря состоит из ключа и связанного с ключом значения. Пример словаря с дубликатами: англо-русский словарь, элементы которого содержат слово на английском языке (ключ) и вариант перевода слова на русский язык (значение), например, {[list] – список, [list] – перечислять}.

Операции словаря:

* создание пустого словаря;
* добавление элемента в словарь;
* исключение элементов с заданным ключом из словаря;
* поиск значений по ключу;
* изменение значения элемента словаря;
* вывод словаря в порядке возрастания ключей.

В демонстрационной программе предусмотреть считывание словаря в начале работы программы из текстового файла и запись словаря в файл перед завершением ее работы.

*2. «Предметный указатель»*

Предметный указатель – это алфавитный список терминов документа с указанием страниц, на которых они упоминаются. Пример элемента указателя: {[заголовочный файл], 108, 189, 927}.

Операции указателя:

* создание пустого предметного указателя;
* добавление термина в предметный указатель;
* редактирование элемента предметного указателя (добавление ссылки на страницу для заданного термина);
* вывод элементов указателя в алфавитном порядке на экран;
* вывод элементов указателя в алфавитном порядке в текстовый файл;
* поиск элемента предметного указателя по термину.

В демонстрационной программе предусмотреть ввод списка терминов с клавиатуры, считывание текста, для которого создается предметный указатель, из текстового файла и запись предметного указателя в другой текстовый файл перед завершением работы программы. Количество строк на странице текста можно задать глобальной константой или вводить с клавиатуры. В текстовом файле нет переноса слов.

*3. «Множество»*

Множество – это совокупность элементов с уникальными значениями, которые, как правило, имеют одинаковый тип данных. Множество должно быть реализовано на основе динамического массива. Реализовать множество таким образом, чтобы можно было легко изменить тип его элементов (использовать шаблоны функций).

Операции множества:

* создание пустого множества;
* включение элемента в множество;
* исключение элемента из множества;
* проверка вхождения элемента в множество;
* объединение двух множеств;
* пересечение двух множеств;
* разность двух множеств;
* проверка равенства двух множеств.

В демонстрационной программе предусмотреть считывание множеств в начале работы программы из текстовых файлов и запись множеств в файлы перед завершением работы программы.

*4. «Словарь»*

Словарь – это индексируемая структура данных, доступ к элементам которой выполняется только по индексу (ключу). Элемент словаря состоит из ключа и связанного с ключом значения. Пример словаря: русско-английский словарь, элементы которого содержат слово на русском языке (ключ) и перевод слова на английский язык, например, {[список], list}. Для хранения словаря в оперативной памяти использовать динамический массив.

Операции словаря:

* создание пустого словаря;
* добавление элемента в словарь;
* исключение элемента из словаря;
* поиск элемента словаря по ключу;
* изменение значения элемента;
* вывод словаря в порядке возрастания ключей.

В демонстрационной программе предусмотреть считывание словаря в начале работы программы из текстового файла и запись словаря в файл перед завершением работы программы. Элементом словаря является абонент телефонной сети (номер телефона, ФИО, паспортные данные). Поле «номер телефона» является ключом. ФИО и паспортные данные являются значением элемента словаря.

*5. «Полином»*

Полином n-степени задается формулой:

a0xn + a1xn-1 + a2xn-2+…+an

Коэффициенты полинома должны храниться в динамическом массиве.

Операции:

* ввод коэффициентов полинома из текстового файла (первое число файла – степень полинома);
* вывод полинома на экран в виде: 4x^3+5x^2-2x^1-6;
* умножение полинома на число;
* изменение знаков коэффициентов полинома на противоположные;
* сложение двух полиномов;
* произведение двух полиномов;
* присваивание одного полинома другому;
* проверка равенства двух полиномов;
* запись полинома в файл: первое число файла – степень полинома, следующие числа - его коэффициенты.

В демонстрационной программе предусмотреть запись выполняемых операций с полиномами в текстовые файлы. Форматы вывода бинарной и унарной операций:

(полином\_1) знак\_бинарной операции (полином\_2) -> результат

знак\_унарной операции (полином) -> результат

*6. «Нижняя треугольная матрица чисел»*

В нижней треугольной матрице элементы, находящиеся выше главной диагонали, равны нулю.

Для хранения значений матрицы для экономии памяти используется одномерный динамический массив, в котором хранятся только элементы из нижней области матрицы. Пример хранения матрицы *A* из четырех строк в одномерном массиве *B* приведен на рис. 3.1.

В демонстрационной программе предусмотреть запись матриц в файлы перед завершением работы программы.

Операции:

* создание нижней треугольной матрицы заданного размера с нулевыми значениями всех элементов;
* установка значения элемента матрицы с индексами *i, j*;
* получение значения элемента матрицы с индексами *i, j*;
* построчный ввод матрицы;
* построчный вывод матрицы;
* сложение двух треугольных матриц;
* проверка равенства двух матриц;
* запись матрицы в файл (первое число файла – размер матрицы);
* чтение матрицы из текстового файла (первое число файла – размер матрицы);
* присваивание одной матрицы другой.

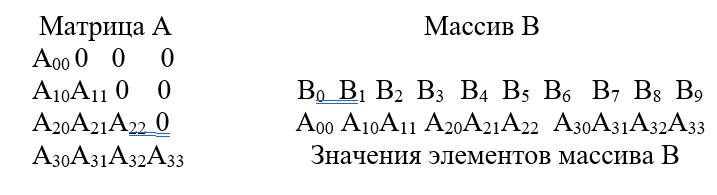


Рисунок 3.1. Хранение нижней треугольной матрицы в одномерном массиве

*7. «Конкурс»*

Программа должна содержать функции для проведения конкурса. Количество участников конкурса и количество призовых мест, на которые претендуют участники, ограничено. Участник конкурса имеет идентификационный номер, имя. Во время проведения конкурса его участникам присваивают баллы.

Операции конкурса:

* открытие конкурса;
* регистрация участника;
* окончание регистрации участников;
* проведение конкурса: регистрация балла, полученного участником конкурса;
* окончание конкурса;
* подведение итогов конкурса: сортировка списка участников по убыванию баллов и вывод упорядоченного списка участников на экран и в текстовый файл.

Регистрация участников может происходить в течение нескольких сеансов работы программы, поэтому при завершении работы программы (до окончания регистрации участников) необходимо участников, зарегистрированных в текущем сеансе, дописывать в конец текстового файла. Регистрация результатов конкурса и подведение итогов выполняется в течение одного сеанса работы программы.

*8. «Числовая матрица с изменяемыми размерами»*

Матрица должна быть представлена в памяти в виде указателя на массив указателей на одномерные динамические массивы чисел.

Операции:

* создание матрицы из данных текстового файла;
* установка значения элемента с заданными индексами (если индексы выходят за границы размеров матрицы, то должно выдаваться сообщение об ошибке);
* получение значения элемента матрицы с заданными индексами (если индексы выходят за границы размеров матрицы, то должно выдаваться сообщение об ошибке);
* изменение количества строк матрицы (если при этом матрица увеличивается, то новые элементы заполняются нулевыми значениями, а если уменьшается, то сохраняются только элементы из верхней левой части матрицы);
* изменение количества столбцов матрицы (если при этом матрица увеличивается, то новые элементы заполняются нулевыми значениями, а если уменьшается, то сохраняются только элементы из верхней левой части матрицы);
* получение текущего количества столбцов;
* получение текущего количества строк;
* вывод матрицы.

В демонстрационной программе предусмотреть считывание матрицы в начале работы программы из текстового файла и запись матрицы в файл перед завершением программы. Первые два числа в файле задают количество строк и столбцов матрицы.

*9. «Множество целых чисел из диапазона 0-n»*

Множество – это совокупность элементов с уникальными значениями, которые, как правило, имеют одинаковый тип данных.

Множество моделируется динамическим массивом из элементов, значения которых могут быть true или false. При включении числа со значением i в множество в элемент массива с индексом i записывается значение true, а при исключении числа со значением i из множества в элемент массива с индексом i записывается значение false.

Операции множества:

* создание пустого множества;
* включение элемента в множество;
* исключение элемента из множества;
* проверка вхождения элемента в множество;
* объединение двух множеств;
* пересечение двух множеств;
* разность двух множеств;
* проверка равенства двух множеств.

Пример множества {1, 3, 4} приведен на рис. 3.2.

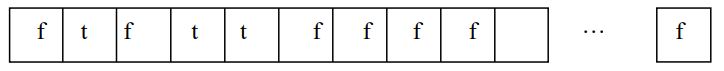


Рисунок 3.2. Хранение множества {1, 3, 4}в массиве

В демонстрационной программе предусмотреть считывание множеств в начале работы программы из текстовых файлов и запись множеств в файлы перед завершением работы программы.

*10. «Ежедневник»*

Ежедневник должен содержать список ежедневных дел и отслеживать своевременность их выполнения.

Операции ежедневника:

* создание пустого ежедневника;
* добавление нового элемента в список дел;
* поиск элементов ежедневника по дате и вывод их, упорядоченных по времени выполнения;
* вывод всего списка элементов ежедневника, упорядоченного по сроку выполнения;
* исключение выполненного элемента из списка дел;
* очистка ежедневника.

В демонстрационной программе предусмотреть считывание существующего списка дел из текстового файла при каждом запуске программы и запись всех невыполненных дел в файл перед завершением работы программы.

*11. «Множество целых чисел из диапазона [0–n] с битовой схемой хранения»*

Множество – это совокупность элементов с уникальными значениями, которые, как правило, имеют одинаковый тип данных.

Для множества используется массив целых чисел (4-х байтных). Каждый элемент множества хранится в виде 1 в соответствующем бите (с номером j) элемента массива с индексом i, что уменьшает размер памяти необходимой множеству. Для выполнения операций с множествами с битовой схемой хранения используются быстрые поразрядные операции.

Операции множества:

* создание пустого множества;
* включение элемента в множество;
* исключение элемента из множества;
* проверка вхождения элемента в множество;
* объединение двух множеств;
* пересечение двух множеств;
* разность двух множеств;
* проверка равенства двух множеств;
* вычисление индекса элемента массива, в котором хранится элемент со значением х (вспомогательная операция);
* вычисление номера разряда элемента массива, в котором хранится элемент со значением х (вспомогательная операция).

В демонстрационной программе предусмотреть запись множеств в текстовые файлы перед завершением программы.

На рис. 3.3 приведена схема хранения множества {1, 30, 32, 64}.

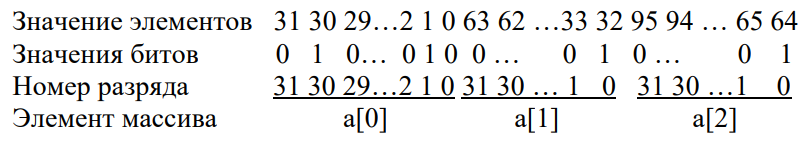


Рисунок 3.3. Схема хранения множества {1, 30, 32, 64}

*12. «Частотный словарь»*

Частотный словарь – это алфавитный список всех слов заданного текста с указанием количества вхождений слова в текст.

Операции словаря:

* создание пустого словаря;
* добавление нового элемента в словарь;
* увеличение значения счетчика слова в элементе словаря с заданным словом;
* вывод элементов словаря в алфавитном порядке;
* поиск элемента словаря по слову.

В демонстрационной программе предусмотреть считывание текста для построения словаря из текстового файла и запись словаря в алфавитном порядке в другой текстовый файл перед завершением работы программы, считывание частотного словаря из текстового файла для просмотра и поиска в нем элементов. В текстовом файле не предусмотрено переноса слов.

*13. «Средства ведения и обработки массива структур»*

В массиве структур каждый элемент имеет признак «занят/свободен». Это позволяет при удалении элемента не перемещать в массиве элементы, а только изменять значение признака.

Операции:

* создание массива структур из данных текстового файла;
* изменение значения элемента с заданным (логическим) номером (логический номер структуры может не совпадать со значением индекса элемента, так как индекс учитывает все элементы массива, в том числе и свободные);
* получение значения элемента матрицы с заданным логическим номером;
* удаление элемента с заданным логическим номером;
* добавление элемента в массив (элемент добавляется в конец массива после последнего занятого элемента, а если последний занятый элемент является последним элементом массива, то перед добавлением нового элемента должно выполняться сжатие массива: исключение всех свободных элементов);
* получение текущего количества занятых элементов;
* получение текущего количества строк;
* вывод массива структур на экран;
* запись занятых элементов структур в текстовый файл.

Поля структуры: номер телефона, ФИО, адрес.

В демонстрационной программе предусмотреть считывание массива структур в начале работы программы из текстового файла и запись массива в файл перед завершением работы программы.

*14. «Верхняя треугольная матрица чисел»*

В верхней треугольной матрице элементы, находящиеся ниже главной диагонали, равны нулю.

Для хранения значений матрицы для экономии памяти используется одномерный динамический массив, в котором хранятся только элементы из верхней области матрицы. Пример хранения матрицы *A* из четырех строк в одномерном массиве *B* приведен на рис. 3.4.

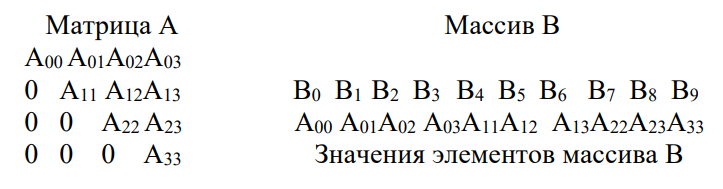


Рисунок 3.4. Хранение верхней треугольной матрицы в одномерном массиве

Операции:

* создание верхней треугольной матрицы заданного размера с нулевыми значениями всех элементов;
* установка значения элемента матрицы с индексами *i, j*;
* получение значения элемента матрицы с индексами *i, j*;
* построчный ввод матрицы;
* построчный вывод на экран матрицы;
* разность двух треугольных матриц;
* проверка равенства двух матриц;
* запись матрицы в файл (первое число файла – размер матрицы);
* чтение матрицы из текстового файла (первое число файла – размер матрицы);
* присваивание одной матрицы другой.

В демонстрационной программе предусмотреть запись матриц в файлы перед завершением программы.

*15. «Предметный указатель»*

Предметный указатель – это алфавитный список терминов документа с указанием страниц, на которых они упоминаются. Пример элемента указателя: {[заголовочный файл], 108, 189, 927}.

Операции указателя:

* создание пустого предметного указателя;
* добавление термина в предметный указатель;
* редактирование элемента предметного указателя (добавление ссылки на страницу для заданного термина);
* вывод элементов указателя в алфавитном порядке на экран;
* вывод элементов указателя в алфавитном порядке в текстовый файл;
* поиск элемента предметного указателя по термину.

В демонстрационной программе предусмотреть ввод списка терминов с клавиатуры, считывание текста, для которого создается предметный указатель, из текстового файла и запись предметного указателя в текстовый файл перед завершением программы. Количество строк на странице текста можно задать глобальной константой или ввести с клавиатуры. В текстовом файле нет переноса слов.

*16. «Перевод целых чисел из одной системы счисления в другую»*

Программа должна предоставлять средства для перевода целых чисел, записанных в системе счисления с основанием q (2<=q<=16) в систему с основанием p (2<=p<=16).

Операции:

* ввод числа-источника и основания его системы счисления с проверкой корректности вводимых данных;
* перевод целого числа из одной системы счисления в другую (при переводе числа из q-ой системы в 10-ую использовать схему Горнера);
* вывод числа, имеющего заданную систему счисления в виде последовательности символов, используемых в системе счисления в основания его системы счисления с проверкой корректности вводимых данных.

*17. «Матричная алгебра»*

Матрица должна быть представлена в памяти в виде указателя на массив указателей на строки матрицы (одномерные динамические массивы).

Операции:

* создание матрицы из чисел текстового файла (первые два числа файла – размеры матрицы);
* построчный вывод матрицы;
* разность двух матриц;
* сумма двух матриц;
* умножение матрицы на число;
* умножение матрицы на вектор;
* умножение двух матриц;
* присваивание одной матрицы другой;
* проверка равенства двух матриц;
* запись матрицы в текстовый файл (первые два числа файла – размеры матрицы).

В демонстрационной программе предусмотреть запись матриц в текстовые файлы перед завершением программы. Реализовать матрицу таким образом, чтобы можно было легко изменить тип ее элементов (использовать шаблоны функций).

*18. «Средства для работы с датами»*

Операции:

* ввод даты в любом формате: дд.мм.гг, дд/мм/гг, дд.мм.гггг, дд/мм/гггг;
* проверка корректности введенной даты;
* вывод даты в длинном формате (10/06/2012);
* вывод даты в американском формате (10/26/2012);
* увеличение даты на 1 день;
* увеличение даты на n дней;
* уменьшение даты на 1 день;
* уменьшение даты на n дней;
* подсчет количества дней между двумя датами;
* вычисление дня недели по дате (например, для даты 12.02.2012 день недели воскресенье).

Вычислить номер дня недели (целое число от 0 до 6, причем 0 соответствует воскресенью) можно по формуле [5]:

Номер\_дня\_недели=Х%7

где Х=abs(int(2.6\*m-0.2)+d+y/4+y+c/4-2\*c),

m – номер месяца,

d – день месяца,

c – номер столетия,

y – номер года в столетии.

В этой формуле март имеет порядковый номер m=1, апрель m=2,…,февраль m=12. Поэтому перед использованием формулы год, месяц (и столетие) даты надо предварительно преобразовать. Например, для даты 1.2.1991 будет m=12, y=1990; для даты 31.12.1991 будет m=10, y=1991.

*19. «Средства для обработки результатов опроса»*

Результаты опроса n респондентов (n<=3000) по m вопросам (m<=30) с двумя альтернативными ответами записаны в текстовом файле. Ответ на вопрос «Да» в файле кодируется 1, а ответ «Нет» - 0. Обработать результаты опроса, используя для хранения данных в оперативной памяти массив из n длинных целых чисел, записывая m ответов одного респондента в m разрядов 32-х разрядного целого числа (*битовая схема хранения)*. Каждый j-ответ i-го респондента хранится в виде 1 или 0 в (i-1) элементе массива в бите с номером j (0 и m+1 разряды числа для m<=30 не используются). Для выполнения операций с данными с битовой схемой хранения используются быстрые поразрядные операции. На рис. 3.5 приведена схема хранения ответов респондентов в оперативной памяти.

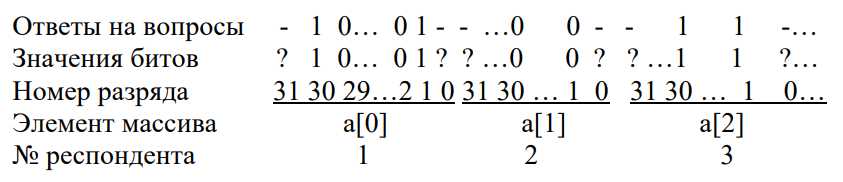


Рисунок 3.5. Схема хранения ответов респондентов {1-респондент {1, 0, …, 0, 1}, 2-респондент {0,…,0}, 3-респондент {1,…1,1},…}

Операции:

* чтение результатов опроса из текстового файла в двоичные разряды массива;
* подсчет ответов «да» и «нет» на j-вопрос;
* поиск вопросов, по которым большинство ответили «да»;
* поиск респондентов, которые на все вопросы ответили «нет»;
* поиск респондентов, которые на все вопросы ответили «да»;
* вывод результатов опроса на экран;
* вычисление индекса элемента массива, в котором хранится элемент со значением х (вспомогательная операция);
* вычисление номера разряда элемента массива, в котором хранится элемент со значением х (вспомогательная операция).

*20. «Нахождение корня нелинейного уравнения»*

Программа должна вычислять корень нелинейного уравнения методами: простых итераций, дихотомии (деления пополам), Ньютона с заданной точностью [6-7]. Программа должна предоставлять пользователю возможность выбора метода решения нелинейного уравнения и задания значения точности вычисления корня. При вычислении корня должно вычисляться количество итераций для достижения заданной точности. Результаты расчета и исходные данные (название метода вычисления и значение точности) в текущем сеансе должны накапливаться в массиве. Перед окончанием работы программы данные из этого массива должны быть добавлены в текстовый файл. Демонстрационная программа должна предоставлять пользователю возможность просмотреть результаты текущего сеанса и результаты, сохраненные в файле.

Операции:

* решение нелинейного уравнения методом итераций;
* решение нелинейного уравнения методом дихотомии;
* решение нелинейного уравнения методом Ньютона;
* вывод на экран результатов текущего сеанса;
* запись в конец текстового файла результатов текущего сеанса;
* просмотр накопленных в текстовом файле результатов.

*21. «Вычисление определенного интеграла»*

Программа должна вычислять приближенное значение определенного интеграла методами: средних, Симпсона, Ньютона-Котеса для m=5 [6‑7]. Программа должна предоставлять пользователю возможность выбора метода вычисления интеграла и задания значения точности его вычисления. При вычислении интеграла должно вычисляться количество итераций для достижения заданной точности. Результаты расчета и исходные данные (название метода вычисления и значение точности) в текущем сеансе должны накапливаться в массиве. Перед окончанием работы программы данные из этого массива должны быть добавлены в текстовый файл. Демонстрационная программа должна предоставлять пользователю возможность просмотреть результаты текущего сеанса и результаты, сохраненные в файле.

Операции:

* вычисление интеграла методом средних;
* вычисление интеграла методом Симпсона;
* вычисление интеграла методом Ньютона-Котеса;
* вывод на экран результатов текущего сеанса;
* запись в конец текстового файла результатов текущего сеанса;
* просмотр накопленных в текстовом файле результатов.