Задания к работе №3 по фундаментальным алгоритмам.

1. Напишите функцию перевода числа из десятичной системы счисления в систему счисления с основанием . При реализации функции разрешается использовать битовые операции и операции обращения к памяти, запрещается использовать стандартные арифметические операции.
2. При реализации функции данного задания запрещается пользоваться строковым представлением числа в двоичной системе счисления. Параметры, фигурирующие в условии, необходимо считать с клавиатуры.

**1.** Написать функцию поиска всех битных целых чисел, в двоичной записи которых присутствует ровно единиц.

**2.** Написать функцию поиска всех битных целых чисел, в двоичной записи которых присутствует ровно подряд идущих единиц.

1. На вход программе через аргументы командной строки подается файл. В файле в каждой строчке содержится информация о сотруднике: id, имя, фамилия, заработная плата. Программа считывает файл, заполняет структуру Employee и в файл трассировки выводит данные, отсортированные по зарплате.
2. Структура Message содержит следующие поля: id сообщения, текст сообщения и длина сообщения в байтах. Программа предлагает пользователю ввести сообщение, после чего происходит заполнение структуры и запись её в файл. Когда пользователь вводит команду, например, ‘stop’, ввод данных завершается, и программа должна считать данные из файла в динамический массив и вывести их на экран.
3. Структура Student содержит поля: id студента, имя, фамилию, группу. Через аргументы командной строки подается файл, содержащий записи о студентах. При старте, программа считывает файл в динамический массив. В программе должен быть реализован поиск по id, по фамилии, по имени, а также вывод студентов по указанной группе.
4. Файл содержит ФИО студента, группу и оценки за 5 экзаменов. Необходимо реализовать структуру Student с советующими полями, затем прочитать файл, заполняя динамический массив структур Student; в трассировочный файл вывести ФИО студента, группу и среднюю оценку за экзамен. Далее необходимо вывести фамилии и имена студентов, чей балл выше среднего за все экзамены.
5. Реализовать структуру String, содержащую в себе поля для динамического массива символов и длины строки. Реализовать функции создания строки, удаления строки, сравнения строк, конкатенации строк, дублирования строки. Продемонстрировать работу с этой структурой.
6. Структура Passenger содержит номер рейса, название аэропорта, из которого вылетает самолет, и город, название аэропорта и город куда прилетает самолет, местное время вылета и местное время прилета (задается структурой). Данные о пассажирах хранятся во входном файле, который задается через аргументы командной строки. Реализовать возможность добавления, просмотра списка пассажиров и удаления пассажиров. Вывод информации о пассажирах должен быть информативным.
7. Структура Mail содержит структуру Address получателя (город, улица, дом, квартира, индекс), вес посылки, почтовый идентификатор. Структура Post содержит структуру Address текущего почтового отделения и динамический массив структур Mail. Реализовать добавление объекта Mail, удаление, информативный вывод при поиске по идентификатору. Объекты структуры Mail должны быть отсортированы по индексу получателя и идентификатору посылки.
8. Через аргументы командной строки подаётся текстовый файл, содержащий последовательность строк, в каждой из которых записаны один или несколько операторов над булевскими векторами из множества [A, B, …, Z] и, возможно, однострочные или многострочные комментарии. Символом однострочного комментария является символ %, а многострочного – символы {}. Возможный вид операторов:
9. A := B <op> C; <op> - логическая операция из списка:

* + (дизъюнкция)
* & (конъюнкция)
* -> (импликация)
* <- (обратная импликация)
* ~ (эквиваленция)
* <> (сложение по модулю 2)
* +> (коимпликация)
* ? (штрих Шеффера)
* ! (стрелка Пирса)

:= - оператор присваивания.

2) X := \W; - логическое отрицание.

3) read(D, base); - ввод значения в переменную D в системе счисления с основанием base.

4) write(Q, base); - вывод значения из переменной Q в системе счисления с основанием base.

Разделителем между операторами является символ “;”. Пробелы, символы табуляций и переносы строк могут присутствовать произвольно, различий между заглавными и строчными буквами нет, вложенные комментарии допускаются, уровень вложенности произвольный. При наличии в командной строке флага /trace, следующим аргументом указывается путь к файлу трассировки (необходимо выводить в файл подробную информацию о выполнении каждой инструкции из файла). Написать программу – интерпретатор инструкций в заданном файле. Необходимо предусмотреть обработку всевозможных ошибок

1. (Детская загадка) Крестьянину нужно перевезти через реку волка, козу и капусту. Но лодка такова, что в ней может поместиться только крестьянин, а с ним или один волк, или одна коза, или одна капуста. Но если оставить волка с козой, то волк съест козу, а если оставить козу с капустой, то коза съест капусту. Как перевезти свой груз крестьянину? Напишите программу, моделирующую поведение крестьянина, а именно исполнение переданных ему команд. В силу сложившихся обстоятельств, крестьянин понимает достаточно ограниченный набор команд:
2. take <object> - взять в лодку заданный объект. Вместо <object> может быть написано wolf, goat или cabbage. При этом команда может быть выполнена, если в лодке есть место.
3. put; - выложить на берег то, что есть в лодке. При этом команда может быть выполнена, если в лодке есть какой-либо объект.
4. move; - переплыть реку на лодке. При этом команда может быть выполнена в любом случае в независимости от того, что именно находится в лодке.

Напишите текстовый файл с последовательностью инструкций для крестьянина, которые решают данную задачу, а также с неверной последовательностью инструкций. Реализуйте программу так, чтобы были обработаны все ошибки и все возможные ситуации, которые могут произойти.

1. На вход программе через аргументы командной строки подаются текстовые файлы с арифметическими выражениями (одно выражение в одной строке). Поступающие файлы имеют произвольное количество записей, выражения в файлах могут быть произвольной структуры: содержать произвольное количество арифметических операций (сложение, вычитание, умножение, целочисленное деление, взятие остатка от деления), круглых скобок и операций возведения в степень. В вашей программе необходимо для каждого выражения, с использованием стека, проверить баланс скобок, вычислить значение каждого выражения, с использованием алгоритма вычисления постфиксного выражения. В результате работы программы на экран выводится имя файла и список арифметических выражения из него с ответами и так для каждого файла. В случае обнаружения ошибки в расстановке скобок для каждого файла, где обнаружена ошибка, необходимо создать текстовый файл, в который помещается ошибочное арифметическое выражение и его номер в исходном файле.