

Типовой расчет (второй семестр) на тему «Моделирование типов. Ссылочные структуры»

Задания по вариантам и группам:

Группы Варианты	А-7-22	А-17-22	А-6-22	А-9-22	А-4-22	А-10-22	А-8-22	А-20-22	Аэ-21-22	А-12-22
1,6,11,16,21,26,31	А	Д	Г	В	Б	Г	В	Б	Г	В
2,7,12,17,22,27,32	Б	А	Д	Г	В	Д	Г	В	Д	Г
3,8,13,18,23,28,33	В	Б	А	Д	Г	А	Д	Г	А	Д
4,9,14,19,24,29,34	Г	В	Б	А	Д	Б	А	Д	Б	А
5,10,15,20,25,30,35	Д	Г	В	Б	А	В	Б	А	В	Б

Требования к отчету:

Часть 1

Задание на Часть 1. (Смоделировать... /Создать новый тип...)

Определение моделируемой структуры данных.

Способ 1

Описание структуры нового типа, схема, программный код.

Описание порядка выполнения всех операций и их программный код.

Способ 2

Описание структуры нового типа, схема, программный код.

Описание порядка выполнения всех операций и их программный код.

В Части 1 описание способов 1 и 2 можно сделать последовательно (один за другим: сначала все операции по способу 1, затем все операции по способу 2), либо для каждой из операций описать сразу оба способа (для попарного сравнения) ее реализации перед описанием следующей операции.

Часть 2

Задание на Часть 2. (На основе описанного нового типа решить...)

Условие задачи (по варианту).

Формат входного текстового файла.

Описание интерфейса (меню) созданного приложения.

Функциональные тесты.

Программный код процедуры для решения задачи (фильтр/поиск)

Заключение

Что было сделано в данной работе.

Сравнение двух способов: недостатки и достоинства (по отношению друг к другу) использованных в работе структур: Списков, Файлов и Массивов, выявленные при моделировании на основе них нового типа.

Вывод: Какой способ оказался проще, удобней? Почему?

Приложения

Приложение 1. Программный код первого модуля (способ 1)

Приложение 2. Программный код второго модуля (способ 2)

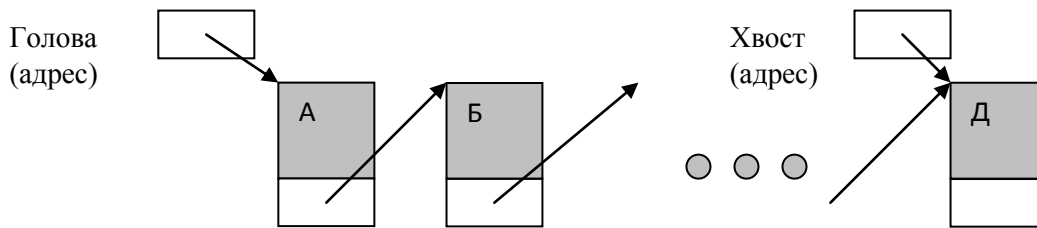
Приложение 3. Программный код приложения, работающего с любым из двух написанных модулей (имена типов и операций должны совпадать в модулях).

Приложение 4. Пример получившегося лог-файла (С выполнением всех возможных действий).

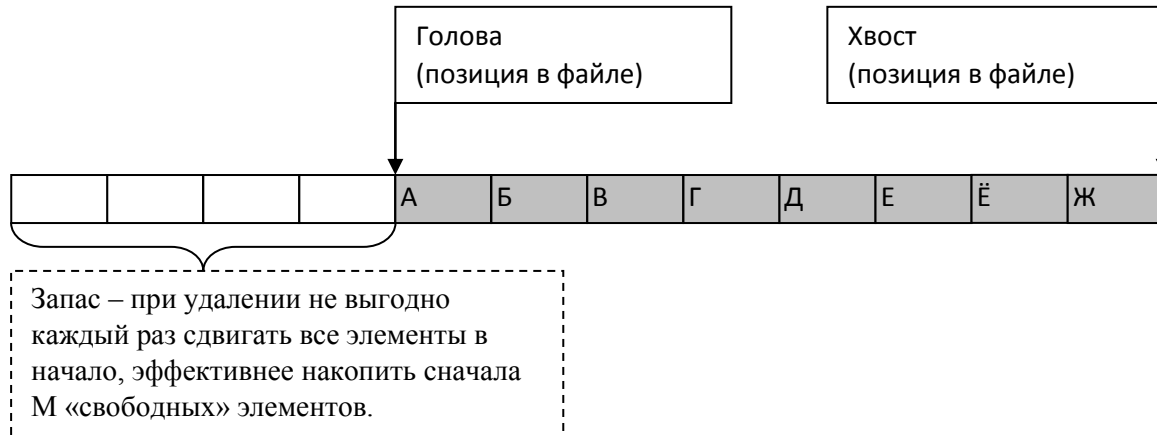
Задание А - «ОЧЕРЕДЬ»

Часть А1. Смоделировать двумя способами новый тип «Очередь»:

1) на основе ссылочного типа «Линейный односвязный список»;



2) на основе типизированного/двоичного файла.



Для этого **создать модуль** и описать **новый тип данных** и его структуру (на русском языке и на Delphi/C, без классов! Только записи/структуры и файловый тип) и **базовые операции**:

- проверка на пустоту очереди;
- создание пустой очереди или очереди из одного элемента (как удобнее);
- добавление элемента в очередь;
- взятие элемента из очереди (считывание + удаление);
- вывод (дописывание) содержимого очереди в текстовый лог-файл для контроля за состоянием очереди после каждой операции с начала и до конца работы программы.

Часть А2. Создать приложение (консольное, на С или Delphi), позволяющее выбрать и выполнить следующие действия над очередью (только с помощью выше перечисленных базовых операций! Без прямого обращения к элементам списка/файла):

- очистить очередь;
- добавить элементы из текстового файла;
- добавить один элемент, на основе данных введенных в интерактивном режиме (с клавиатуры или с формы);
- обработать (удалить) N элементов очереди; (см. пример в Задании Г про Дек)
- фильтр: удалить из очереди все элементы, не удовлетворяющие условию поиска, указанному в списке задач по вариантам (стр.3).

После каждой операции выводить содержимое очереди в лог-файл.

При добавлении элемента предусмотреть его *коррекцию* в соответствии с условием для коррекции из списка задач по вариантам (стр.3).

Список задач для задания А – «ОЧЕРЕДЬ», Б – «СПИСОК» и Г – «ДЕК»

1. *Исходные данные:* Багаж авиапассажира характеризуется фамилией пассажира, номером рейса, датой вылета, количеством вещей и общим весом вещей. *Условие поиска:* Пассажиры, багаж которых состоит из одной вещи весом не менее 30 кг, и вычислить средний вес багажа всех пассажиров. *Коррекция:* Вес и количество вещей неотрицательные и не более 50. При нулевом количестве вещей только нулевой вес.
2. *Исходные данные:* Расписание экзаменов содержит следующую информацию: номер студенческой группы, дата и время экзамена, номер аудитории, наименование дисциплины и фамилия преподавателя. *Условие поиска:* Даты, по которым занят конкретный преподаватель (Запросить фамилию в режиме диалога). *Коррекция:* Номер месяца 1-12, дней не более 28-31 в соответствии с месяцем.
3. *Исходные данные:* Сведения о автомобиле состоят из его марки, номера, фамилии владельца и отметки о прохождении техосмотра. *Условие поиска:* Фамилии владельцев и номера автомобилей определенной марки (Запросить марку в режиме диалога), не прошедших вовремя техосмотр. *Коррекция:* Отметка о прохождении техосмотра включает и дату его прохождения. Техосмотр должен быть пройден не более чем год назад. Номер месяца 1-12, дней не более 28-31 в соответствии с месяцем.
4. *Исходные данные:* В расписании поездов указаны: номер поезда, пункт отправления, пункт назначения, время отправления, время в пути, цена билета. *Условие поиска:* Поезда, следующие из Москвы в Санкт-Петербург, время отправления которых с 7.00 до 9.00. *Коррекция:* Время отправления: минуты 0-59, часы 0-23. Названия пунктов отправления и назначения могут состоять из нескольких слов.
5. *Исходные данные:* В анкетах сотрудников некоторого учреждения имеются следующие сведения: фамилия, имя, отчество, год рождения, должность (мастер, инженер, рабочий и т..д.), стаж, образование. *Условие поиска:* Мастера с высшим образованием, и вычислить их средний стаж. *Коррекция:* Стаж 0-60 лет, и не больше возраста-15лет.
6. *Исходные данные:* В сведениях об игрушках указывается название (кукла, кубики, мяч, конструктор и т.д.), цена, возрастные границы ребенка (например, от двух до пяти лет). *Условие поиска:* Игрушки, цена которых не превышает заданную (в режиме диалога) и которые подходят детям пяти лет. *Коррекция:* Возрастные границы: нижняя меньше верхней не менее чем на 2 года, не отрицательные и не более 150 лет.
7. *Исходные данные:* Имеются следующие сведения о книгах: фамилия и инициалы автора, название, год издания. *Условие поиска:* Книги, изданные с 1990 г., в названии которых содержится слово «информатика» в любом регистре. *Коррекция:* Год издания не менее 1800, и не более нынешнего года.
8. *Исходные данные:* О каждой книге имеются следующие сведения: фамилия и инициалы автора, название, год издания и количество запросов за последние три года. *Условие поиска:* Книги, изданные до 1980 года, по которым нет запросов *Коррекция:* Год издания не менее 1800, и не более нынешнего года. Запросов неотрицательное количество.
9. *Исходные данные:* Сведения состоят из фамилий, имен и отчеств сотрудников и их телефонов. *Условие поиска:* Сотрудники, номера телефонов которых начинаются с цифр 361, 362 или 273. *Коррекция:* Номера телефонов только из цифр, и их не менее 6.
10. *Исходные данные:* Имеются сведения о химических веществах: название, удельный вес, проводимость (проводник, полупроводник, изолятор), количество. *Условие поиска:* Полупроводники, имеющиеся в количестве более 5 кг. *Коррекция:* Проводимость только трех указанных видов, количество положительное.
11. *Исходные данные:* Сведения о компьютерах некоторого учреждения состоят из марки компьютера, даты приобретения, номера комнаты, где находится компьютер, и фамилии ответственного лица. *Условие поиска:* Компьютеры IBM PC-486, приобретенные в 1995 г. *Коррекция:* В дате номер месяца 1-12, дней не более 28-31 в соответствии с месяцем, номер комнаты положительный.

12. *Исходные данные:* Сведения об изделиях, хранящихся на складе, состоят из порядкового номера, наименования и количества. *Условие поиска:* Изделия, которых меньше заданного (в режиме диалога) количестве. *Коррекция:* Количество, имеющееся в наличии и заданное, - положительные величины.
13. *Исходные данные:* Имеются сведения об экспортируемых товарах: наименование, страна-экспортер, срок поставки и количество товара. *Условие поиска:* Товары, срок поставки которых - до 01 июля текущего года. *Коррекция:* Номер месяца 1-12, дней не более 28-31 в соответствии с месяцем.
14. *Исходные данные:* Сведения о заказах некоторого предприятия состоят из номера заказа и даты (число, месяц, год) его исполнения. *Условие поиска:* Все весенние заказы. *Коррекция:* Номер заказа положительный, номер месяца 1-12, дней не более 28-31 в соответствии с месяцем.
15. *Исходные данные:* Некоторое обслуживающее предприятие хранит сведения о заказах: наименование заказа и время (часы, минуты, секунды) его поступления. *Условие поиска:* Заказы, которые произошли с 15ч 20 мин 35 с до 18ч 50 мин 10с. *Коррекция:* Время поступления: часы 0-23, минуты и секунды 0-59.
16. *Исходные данные:* Сведения о заказе состоят из адреса (улица, дом, квартира) и даты (число, месяц, год) исполнения. *Условие поиска:* Заказы, которые должны были быть выполнены 15 мая 2015 г на Московской улице. *Коррекция:* Номер месяца 1-12, дней не более 28-31 в соответствии с месяцем. Номер квартиры положительное число.
17. *Исходные данные:* Сведения о горных вершинах состоят из следующих данных: название, координаты (широта и долгота), высота (в метрах над уровнем моря). *Условие поиска:* Вершины с высотой более 3000м, расположенные в Северном полушарии. *Коррекция:* Широта от -90 до 90, долгота от -180 до 180. Высота положительная.
18. *Исходные данные:* По некоторой группе людей собраны следующие сведения данные: возраст, пол, рост, вес, температура, артериальное давление (верхнее и нижнее). *Условие поиска:* Мужчины моложе 25 лет, имеющих верхнее артериальное давление больше 140. *Коррекция:* Пол только мужской и женский, некорректный сделать мужским. Давление положительное.
19. *Исходные данные:* О детях, поступающих в спортивную секцию, собраны следующие данные: фамилия, имя, год рождения, вес, рост. *Условие поиска:* Дети от 10 до 12 лет, рост которых не менее 155см, а вес не более 45 кг. *Коррекция:* Все дети не младше 5 лет и не старше 14 по году рождения, рост-вес положительные и не более 200.
20. *Исходные данные:* Электронная записная книжка должна содержать следующие сведения: фамилия, имя, отчество, адрес (город, улица, дом, квартира), телефон. *Условие поиска:* Всех, проживающих в г.Саратове на улице Солнечной. *Коррекция:* Дом, квартира - положительные числа.
21. *Исходные данные:* Имеются сведения о районных библиотеках города: номер, тип (взрослая, детская), район, адрес, часы работы. *Условие поиска:* Детские библиотеки заданного (в режиме диалога) районе. *Коррекция:* Часы работы: минуты 0-59, часы 6-21. Начало работы раньше завершения не менее чем на 4 часа.
22. *Исходные данные:* Сведения о магазинах города состоят из названия, типа (книжный, продуктовый, универсам и т.д.), района, адреса и часов работы. *Условие поиска:* Книжные магазины заданного (в режиме диалога) района, работающие после 18 часов, и число таких магазинов. *Коррекция:* Начало работы раньше завершения не менее чем на 4 часа. Часы работы: минуты 0-59, часы 0-23.
23. *Исходные данные:* Имеются следующие данные о товарах обувного магазина: наименование (ботинки, сапоги, босоножки и т.д.), тип (мужская, женская или детская), сезон, цвет, размеры, цена. *Условие поиска:* Летняя детская обувь и ее средняя цена. *Коррекция:* Размеры с 10 до 45, типов только три.

24. *Исходные данные:* Имеются сведения о кошках, участвующих в конкурсе: порода, кличка, окрас, возраст, Фамилия хозяина, место в конкурсе. *Условие поиска:* Кошки сиамской породы, вошедшие в призовую десятку, и их средний возраст. *Коррекция:* Возраст от 0 до 15, место положительное.

25. *Исходные данные:* Сведения о маршрутах городских автобусов содержат: номер маршрута, начальный пункт, конечный пункт, описание маршрута (промежуточные остановки), среднее время движения по маршруту, интервал в час пик. *Условие поиска:* Маршруты с заданным (в режиме диалога) конечным пунктом и интервалом в часы пик, меньшим 10 мин, и число таких маршрутов. *Коррекция:* Интервалы положительные и не более часа, среднее время не более двух часов и не менее 30 минут.

26. *Исходные данные:* Имеются сведения о пациентах поликлиники: фамилия, имя, отчество, год рождения, номер полиса ОМС, адрес, основное заболевание, дата последнего посещения лечащего врача. *Условие поиска:* Больные диабетом, не посещавшие лечащего врача более трех месяцев, и общее число больных диабетом. *Коррекция:* В дате посещения номер месяца 1-12, дней не более 28-31 в соответствии с месяцем.

27. *Исходные данные:* Имеются сведения о прохождении детьми прививок: фамилия, имя, год рождения, адрес, отметки о прививках против туберкулеза, полиомиелита, кори, коклюша, столбняка. *Условие поиска:* Дети семи лет, у которых отсутствует хотя бы одна прививка. *Коррекция:* В дате рождения номер месяца 1-12, дней не более 28-31 в соответствии с месяцем. Отметки только двух видов: сделана по плану или не сделана.

28. *Исходные данные:* Сведения о товарах магазина одежды: наименование (костюм, пальто и т.д.), размер, рост, цвет, номер модели. *Условие поиска:* Костюмы, начиная с 52 размера. *Коррекция:* Размеры с 20 по 60, рост 100-200.

29. *Исходные данные:* Имеются сведения о выпускниках учебного заведения: фамилия, имя, отчество, год поступления, год окончания, специальность, отметка о работе о специальности. *Условие поиска:* Выпускники, окончившие учебное заведение в течение последних трёх лет и не работающие по специальности. *Коррекция:* Год окончания больше года поступления не менее чем на 4 года, но не больше нынешнего.

30. *Исходные данные:* Имеются сведения о программных продуктах: наименование, автор, соавторы, версия, год создания. *Условие поиска:* Вывести сведения об авторах, создавших программный продукт без соавторов в заданном (в режиме диалога) году. *Коррекция:* Год создания и заданный год не раньше 1950 и не больше нынешнего.

31. *Исходные данные:* Расписание занятий содержит следующую информацию: день недели, четность недели, номер пары, название студенческой группы, номер аудитории. *Условие поиска:* Все занятия, проходящие по нечетным неделям в заданной аудитории (Запросить аудиторию в режиме диалога). *Коррекция:* Дни недели только понедельник-суббота, номера пар 1-6.

32. *Исходные данные:* Сведения о городах состоят из следующих данных: названия города и страны, координаты (широта и долгота), численность населения. *Условие поиска:* Города с численностью населения более 1 млн, расположенные в Южном полушарии. *Коррекция:* Широта от -90 до 90. Численность населения положительная.

33. *Исходные данные:* Имеются сведения о магазинах города: район, адрес, часы работы. *Условие поиска:* Найти все магазины, работающие 24 часа (с 0:00 до 24:00) заданного (в режиме диалога) района. *Коррекция:* Часы работы: минуты 0-59, часы 0-24. У 24 часов только 0 минут, продолжительность работы не менее 6 часов.

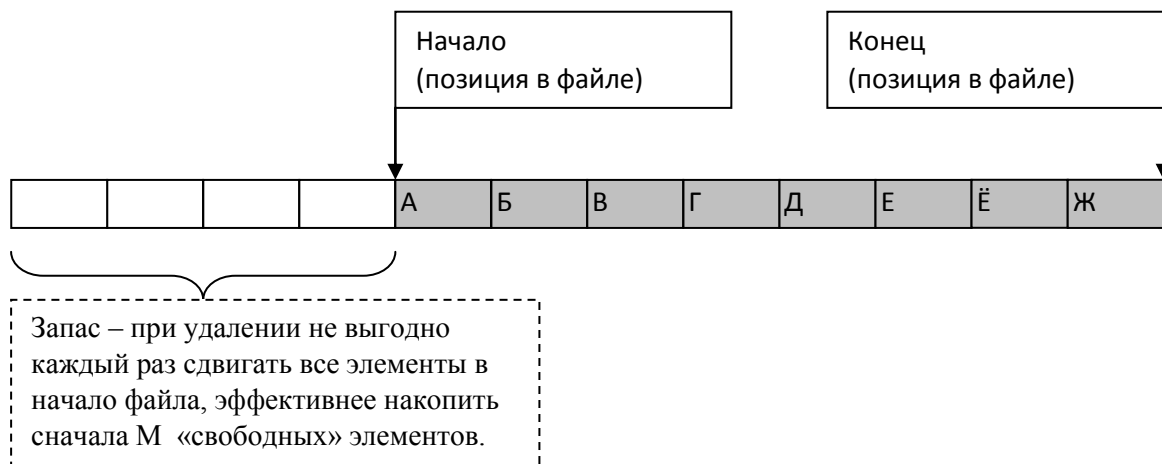
Задание Б – «СПИСОК»

Часть Б1. Смоделировать двумя способами новый тип «Список»:

1) на основе одномерного динамического массива;



2) на основе типизированного/двоичного файла.



Для этого **создать модуль** и описать **новый тип данных** и его структуру (на русском языке и на Delphi/C, без классов! Только записи/структуры, массивы и файловый тип) и **базовые операции**:

- проверка на пустоту списка;
- создание пустого списка или списка из одного элемента (как удобнее);
- добавление элемента в список (в начало/в конец/после текущего);
- удаление элемента из списка (из начала/из конца/после текущего);
- сделать текущим элементом в списке (первый/ последний/ следующий/ предыдущий);
- считать значение текущего элемента из списка;
- вывод (дописывание) содержимого списка в текстовый лог-файл для контроля за состоянием списка после каждой операции с начала и до конца работы программы.

Часть Б2. Создать приложение (консольное, на C или Delphi), позволяющее выбрать и выполнить следующие действия над списком (только с помощью выше перечисленных базовых операций! Без прямого обращения к элементам массива/файла):

- очистить список;
- добавить элементы из текстового файла (в начало/ в конец/ после текущего);

- добавить один элемент, на основе данных введенных в интерактивном режиме (с клавиатуры или с формы) (в начало/ в конец/ после текущего);
- удалить N элементов списка (из начала/ из конца/ после текущего);
- сделать текущим элементом первый/последний/предыдущий/следующий;
- фильтр: *удалить* из списка все элементы, не удовлетворяющие *условию поиска*, указанному в списке задач по вариантам (стр.3), и *продублировать* удовлетворяющие условию.

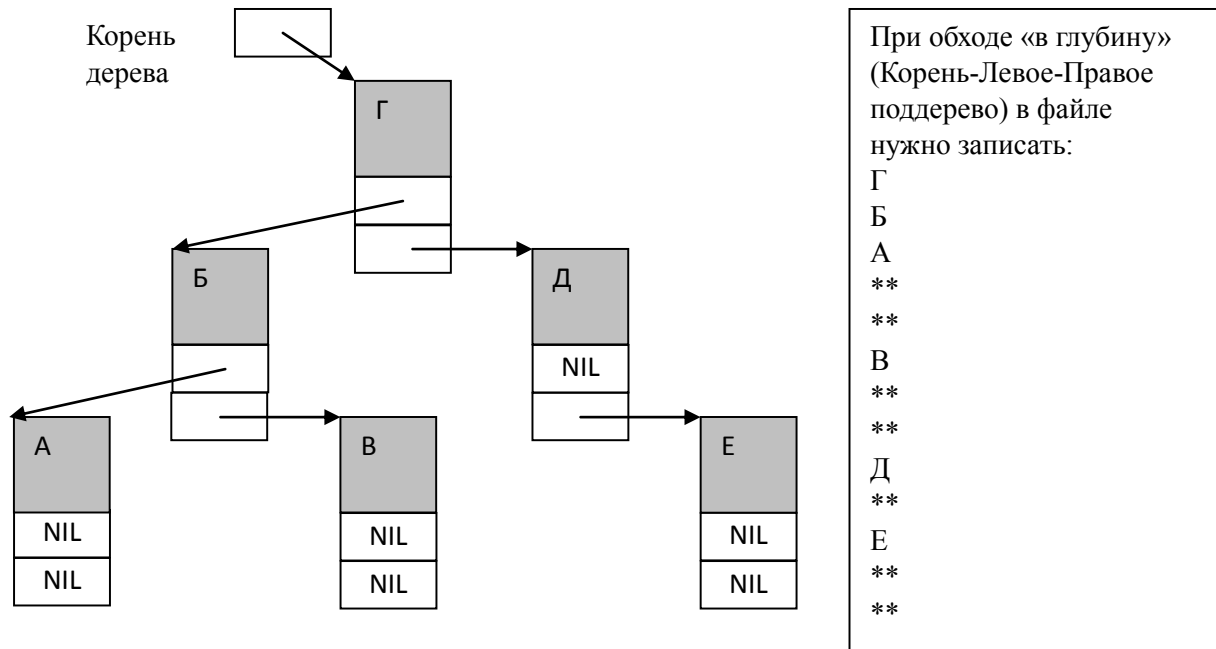
После каждой операции выводить содержимое списка в лог-файл.

При *добавлении* элемента предусмотреть его *коррекцию* в соответствии с условием *коррекции* из списка задач по вариантам (стр.3).

Задание В – «БИНАРНОЕ ДЕРЕВО»

Часть В1. Смоделировать двумя способами новый тип «Бинарное дерево»:

1) на основе ссылочного типа «Нелинейный связный список»;

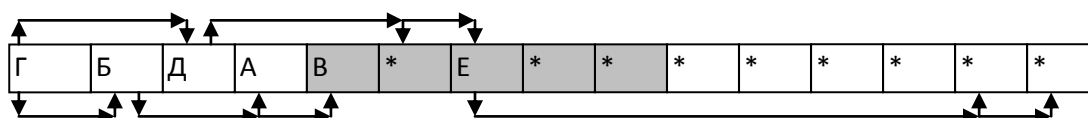


При выводе Левое_поддерево – Корень – Правое_поддерево получим:

АБ**В**Г**Д**Е** (** - пустая вершина)

(без вывода пустых вершин) получим: АБВГДЕ

2) на основе динамического массива или типизированного/двоичного файла (на выбор студента).



Номера дочерних вершин рассчитываются по формулам

$2N$ и $2N+1$ для родительских вершин с номерами $N = 1, 2, 3, 4, \dots$

При расчете индексов/позиций в файле с 0 нужна поправка +1 к индексу родительской вершины

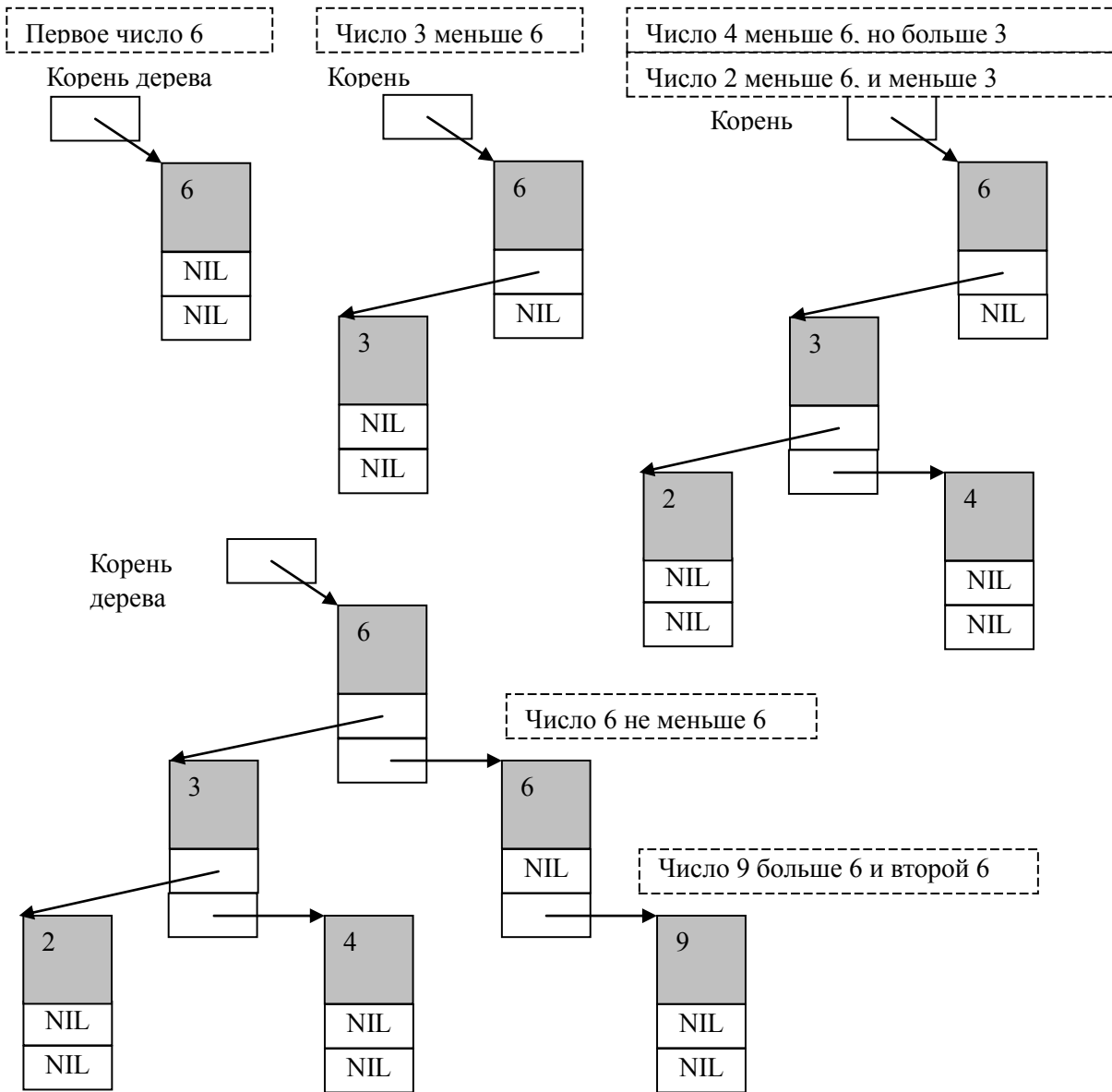
Для этого **создать модуль** и описать **новый тип данных** и его структуру (на русском языке и на Delphi/C, без классов! Только записи/структуры и массив/файл) и **базовые операции**:

- проверка на пустоту дерева (поддерева, начинающегося с текущей вершины);
- создание пустого дерева или дерева из одного элемента (как удобнее);
- сделать текущим (по адресу в списке или индексу в массиве) корень дерева;
- сделать текущим левое поддерево (от текущей вершины);
- сделать текущим правое поддерево (от текущей вершины);
- сделать текущим родительскую вершину для текущей вершины (для удобства можно в способе 1 ввести третью связь – обратную связь с родительской вершиной, чтобы не искать ее по всему дереву – Трёхсвязный список);
- узнать значение текущей вершины;
- изменить значение текущей вершины;
- создать левое поддерево для текущей вершины;
- создать правое поддерево для текущей вершины;

Часть В2. Создать приложение (консольное, на C или Delphi), позволяющее выбрать и выполнить следующие действия над деревом (только с помощью выше перечисленных базовых операций! Без прямого обращения к элементам списка/массива/файла):

- удалить дерево (освободить память);
- создать упорядоченное по ключу (по возрастанию/убыванию одного или нескольких полей записи/структуры) бинарное дерево из текстового файла;
- создать неупорядоченное бинарное дерево (генеалогическое дерево) из текстового файла (Корень – Левое_поддерево – Правое_поддерево, ** – признак пустого поддерева);
- добавить один элемент, со значениями введенными в интерактивном режиме (с клавиатуры) в упорядоченное дерево по ключу (по возрастанию/убыванию одного или нескольких полей записи/структуры) для построения упорядоченного дерева; например, в ниже изображенное дерево добавляются числа 6,3,4,2,6,9 по возрастанию (неубыванию в случае добавления одинаковых значений);
- найти решение задачи из списка заданий по вариантам в конце этого файла (стр.16);
- вывести дерево 2-4мя разными способами: (Корень – Левое_поддерево – Правое_поддерево, Левое_поддерево – Корень – Правое_поддерево, Левое_поддерево – Правое_поддерево – Корень; графически(по желанию). ** – признак пустого поддерева);

После каждой операции выводить содержимое дерева в текстовый лог-файл для контроля за состоянием бинарного дерева после каждой операции с начала и до конца работы программы первым или последним из вышеуказанных 4-х способов.

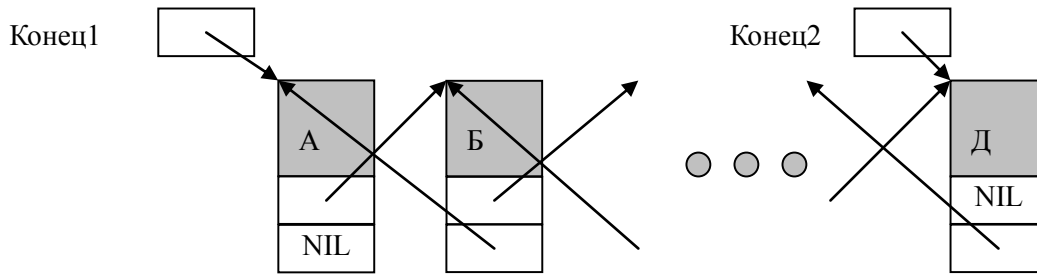


При выводе Левое_поддерево – Корень – Правое_поддерево получим возрастающую(неубывающую) последовательность: 2,3,4,6,6,9

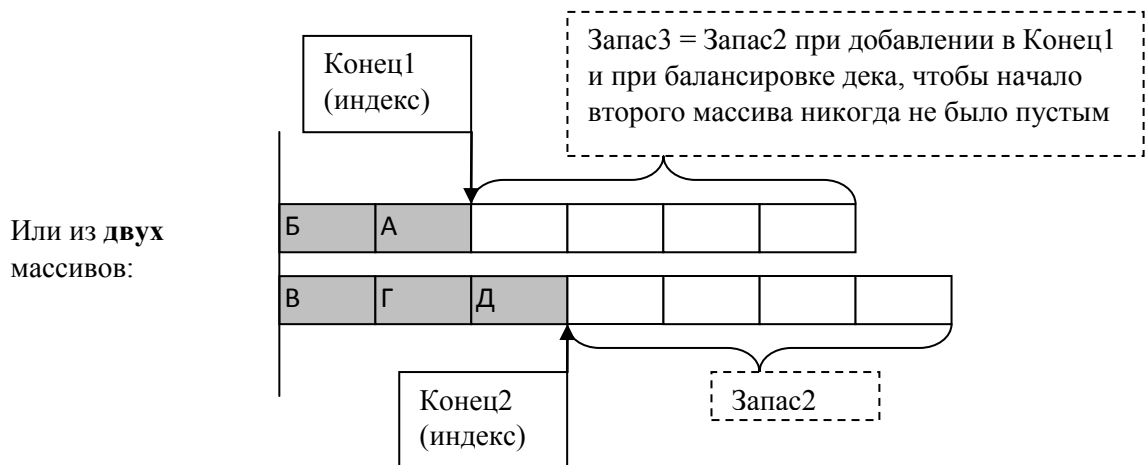
Задание Г – «ДЕК»

Часть Г1. Смоделировать двумя способами новый тип «Дек» (DEQ = Double Ended Queue = «Двухконечная очередь»):

1) на основе ссылочного типа «Линейный двусвязный список»;



2) на основе динамического массива(ов).



Для этого **создать модуль** и описать **новый тип данных** и его структуру (на русском языке и на Delphi/C, без классов! Только записи/структуры и массив(ы)) и **базовые операции**:

- проверка на пустоту дека;
- создание пустого дека или дека из одного элемента (как удобнее);
- добавление элемента в дек (в конец, играющего в данный момент роль «хвоста»);
- взятие элемента из дека (считывание + удаление из конца, играющего в данный момент роль «головы»);
- поменять направление (поменять «роль» обоих концов: «голова» \leftrightarrow «хвост»);
- вывод (дописывание) содержимого дека (с текущего конца-головы) в текстовый лог-файл для контроля за состоянием дека после каждой операции с начала и до конца работы программы.

Часть Г2. Создать приложение (консольное, на С или Delphi), позволяющее выбрать и выполнить следующие действия над деком (только с помощью выше перечисленных базовых операций! Без прямого обращения к элементам списка/массива):

- очистить дек (освободить память);
- добавить элементы из текстового файла;
- добавить один элемент, на основе данных введенных в интерактивном режиме (с клавиатуры);
- обработать (удалить) N элементов дека; например:

```
i:=0; // подсчет удаленных
While not IsEmpty(Deq) and (i<N) do // если дек не пуст и не все N удалены
Begin
    info:=GetFirst(Deq); inc(i); //изъять первый элемент из тек.головой и увеличить счетчик i
End;
```

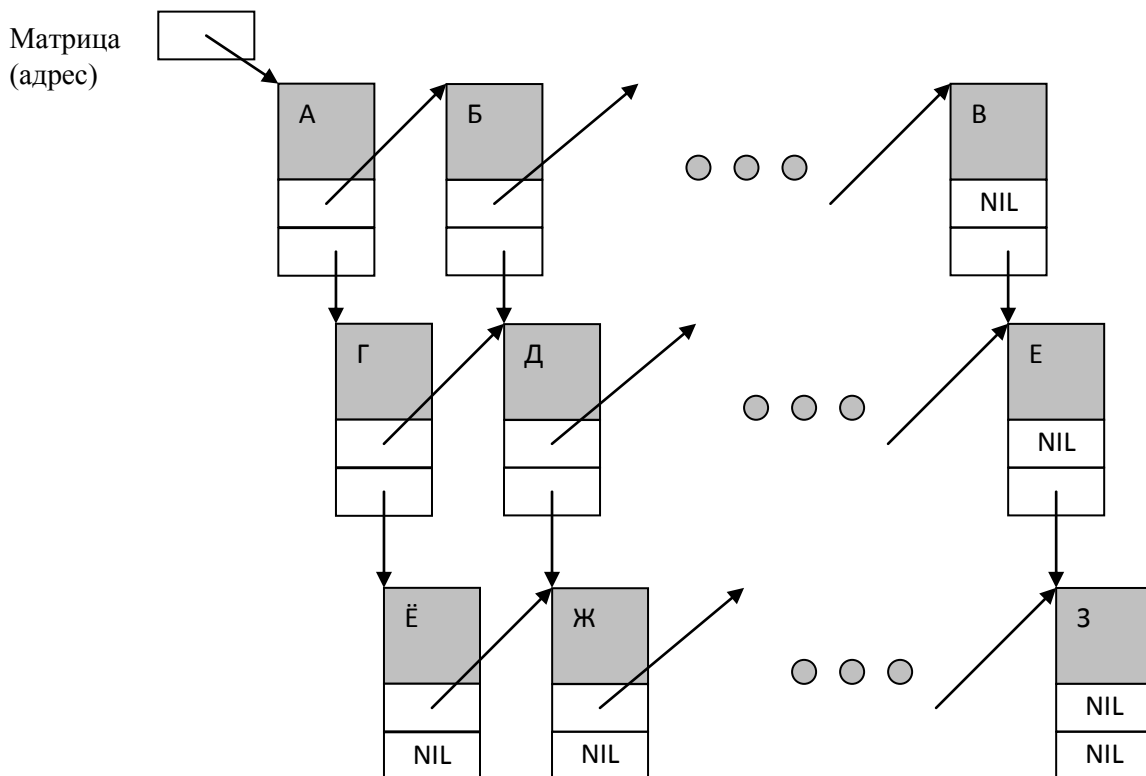
- поменять направление (поменять «роль» обоих концов: «голова» \leftrightarrow «хвост»);
- фильтр: разделить дек на два: с элементами удовлетворяющими и с неудовлетворяющими условию поиска, указанному в списке задач по вариантам (стр.3).

Все операции выполнять с текущего конца-головой и после каждой операции выводить содержимое дека в лог-файл (тоже начиная с текущего конца-головой), при добавлении элемента предусмотреть его *коррекцию* в соответствии с условием коррекции из списка задач по вариантам (стр.3).

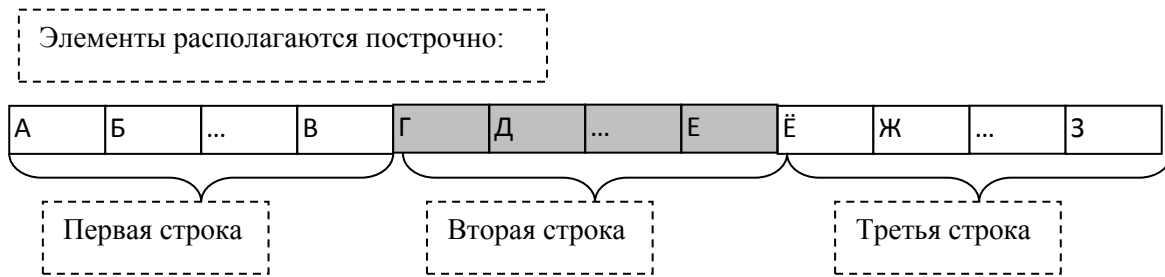
Задание Д – «МАТРИЦА»

Часть Д1. Смоделировать двумя способами новый тип «Матрица»:

- 1) на основе ссылочного типа «Нелинейный двусвязный список»;



2) на основе типизированного/двоичного файла.



Для этого **создать модуль** и описать **новый тип данных** и его структуру (на русском языке и на Delphi/C, без классов! Только записи/структуры и массив(ы)) и **базовые операции**:

- проверка на пустоту массива;
 - создание пустого массива размера NxM (выделение памяти);
 - добавление строки после последней;
 - добавление столбца после последнего;
 - очистить матрицу (освободить память);
 - сделать текущим элементом первый элемент i-й строки;
 - сделать текущим элементом первый элемент j-го столбца;
 - сделать текущим следующий элемент в строке/столбце;
 - чтение значения текущего элемента (без изменения текущей позиции);
 - изменение значения текущего элемента (без изменения текущей позиции);
 - * изменение элемента по индексам (строки и столбца) (без изменения текущей позиции);
 - * чтение элемента по индексам (строки и столбца) (без изменения текущей позиции);
 - вывод (дописывание) содержимого матрицы в текстовый лог-файл для контроля за состоянием матрицы после каждой операции во время работы программы.
- (-* - обратите внимание: частое использование этих операций будет неэффективным для первого способа, но для обоих способов нужно реализовать *полный* список операций).

Часть Д2. Создать приложение (консольное, на C или Delphi), позволяющее выбрать и выполнить следующие действия над матрицей (только с помощью выше перечисленных базовых операций! Без прямого обращения к элементам списка/файла):

- очистить матрицу (освободить память);
- создать матрицу, считав размеры и значения элементов из текстового файла;

Например, для открытого для чтения текстового файла *f* код может быть таким:

```

Readln(f, n, m); // ввод размеров из файла
Alloc_mem(a, n, m); // выделение памяти для A(nxm), и обнуление при желании
For i:=1 to n do // для каждой строки
Begin
    goto_line(a, i); // сделать текущим 1-й элемент i-ой строки
    For j:=1 to m do // m раз выполнить:
    Begin
        Read(f, value); // считать значение (value) элемента (Aij) из файла
        Change_current(a, value); // изменить значение текущего
        Goto_next_j(a); // сделать текущим следующий в строке
    End;
    Readln(f); // переход на следующую строку в файле
End;

```

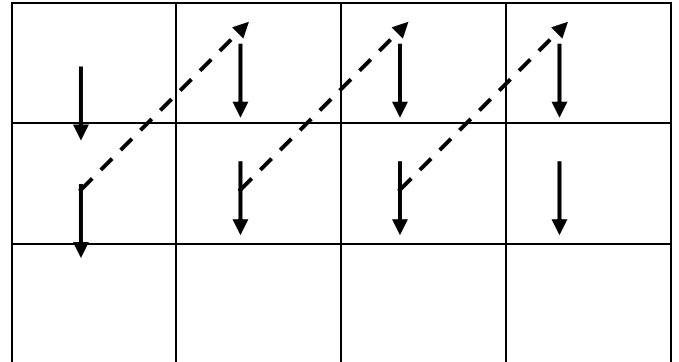
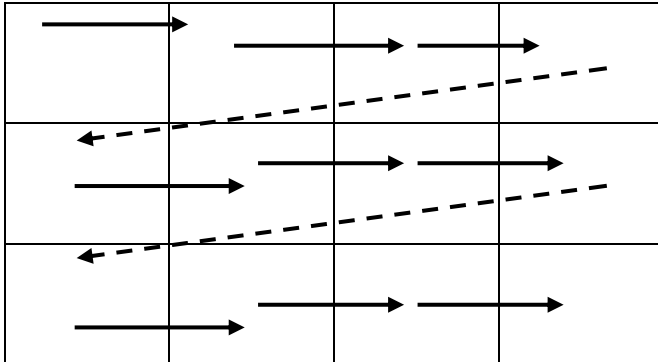
- изменить один элемент по индексам, на основе данных введенных в интерактивном режиме (с клавиатуры или с формы);
 - изменить размер матрицы, добавив строку или столбец из 0(нулей);
 - поиск: решить задачу из нижеприведенного списка задач для Задания Д.
- После каждой операции выводить содержимое матрицы в лог-файл.

Список задач для Задания Д – «Матрица»:

Порядок обхода элементов матрицы:

По строкам – четные варианты

По столбцам – нечетные варианты



1. Задан целочисленный двумерный массив A из n строк и m столбцов. Найти номер последнего максимального элемента среди положительных элементов, начиная с первого элемента, большего T . Матрицу рассматривать по столбцам. В случае отсутствия положительных или больших T элементов или при невозможности поиска вывести соответствующие поясняющие сообщения.
2. Задан целочисленный двумерный массив A из n строк и m столбцов. Найти минимальное значение среди элементов, меньших B , расположенных до первого элемента, большего A_{11} . Матрицу рассматривать по строкам. В случае отсутствия элементов, меньших B или больших A_{11} , или невозможности поиска вывести соответствующие поясняющие сообщения.
3. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер первого максимального элемента среди отрицательных элементов, расположенных до первого элемента, большего T . Матрицу рассматривать по столбцам. В случае отсутствия отрицательных или больших T элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
4. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти максимальное значение среди отрицательных элементов, расположенных до первого элемента, равного T . Матрицу рассматривать по строкам. В случае отсутствия отрицательных или равных T элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
5. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти максимальное значение среди отрицательных элементов, расположенных до первого элемента, меньшего X . Матрицу рассматривать по столбцам. В случае отсутствия отрицательных или меньших X элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
6. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер последнего максимального значения среди отрицательных элементов, расположенных после первого элемента, равного T . Матрицу рассматривать по строкам. В случае отсутствия отрицательных или равных T элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
7. Задан целочисленный двумерный массив A из n строк и m столбцов. Найти номер последнего минимального элемента среди элементов, меньших $T1$ и расположенных до первого элемента, большего $T2$. Матрицу рассматривать по столбцам. В случае отсутствия меньших $T1$ или больших $T2$ элементов или невозможности поиска вывести соответствующие поясняющие сообщения.

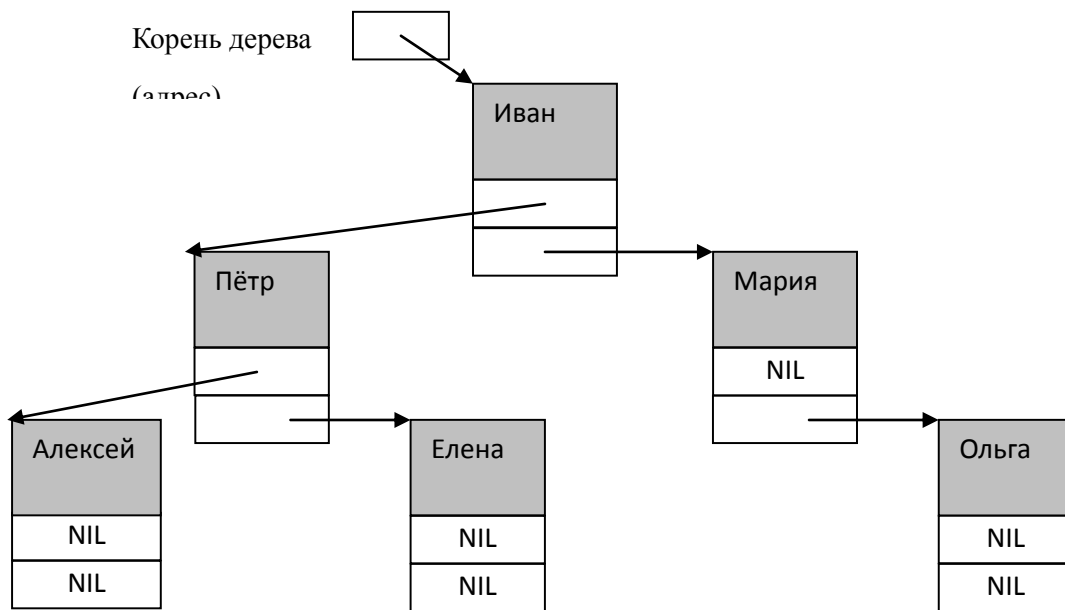
8. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти значение максимального элемента среди четных (по значению) элементов, расположенных до первого нечетного элемента. Матрицу рассматривать по строкам. В случае отсутствия четных или нечетных элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
9. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер первого минимального элемента среди элементов, больших $T1$ и расположенных после первого элемента, равного $T2$. Матрицу рассматривать по столбцам. В случае отсутствия больших $T1$ или равных $T2$ элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
10. Задан целочисленный двумерный массив A из n строк и m столбцов. Найти номер последнего максимального элемента среди элементов, значения которых лежат в диапазоне $[c,d]$ ($c \leq A_i \leq d$) и расположенных до первого четного элемента. Матрицу рассматривать по строкам. В случае отсутствия четных элементов или элементов, значения которых лежат в указанном диапазоне, или невозможности поиска вывести соответствующие поясняющие сообщения.
11. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер последнего минимального элемента среди четных положительных элементов, лежащих после первого отрицательного элемента. Матрицу рассматривать по столбцам. В случае отсутствия отрицательных или четных положительных элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
12. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер последнего максимального элемента среди элементов, меньших $T1$ и лежащих после первого элемента, равного $T2$. Матрицу рассматривать по строкам. В случае отсутствия меньших $T1$ или равных $T2$ элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
13. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер первого максимального элемента среди элементов, значения которых лежат в диапазоне от T до D (не включительно) и расположенных после первого положительного элемента. Матрицу рассматривать по столбцам. В случае отсутствия элементов в указанном диапазоне значений или положительных элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
14. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер первого максимального значения среди отрицательных элементов, расположенных до первого элемента, равного T . Матрицу рассматривать по строкам. В случае отсутствия отрицательных или равных T элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
15. Задан целочисленный двумерный массив A из n строк и m столбцов. Найти минимальное значение положительных элементов, расположенных после первого элемента, кратного трем. Матрицу рассматривать по столбцам. В случае отсутствия положительных или кратных трем элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
16. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер первого минимального значения среди положительных элементов, расположенных после первого элемента, равного нулю. Матрицу рассматривать по строкам. В случае отсутствия положительных или нулевых элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
17. Задан целочисленный двумерный массив A из n строк и m столбцов. Найти значение максимального элемента среди элементов, кратных $k1$, и расположенных до первого отрицательного элемента. Матрицу рассматривать по столбцам. В случае отсутствия кратных $k1$ или отрицательных элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
18. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер первого минимального элемента среди положительных элементов, расположенных до первого элемента, кратного пяти. Матрицу рассматривать по строкам. В случае отсутствия положительных или кратных пяти элементов или невозможности поиска вывести соответствующие поясняющие сообщения.

19. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти минимальное значение положительных элементов, расположенных после первого элемента, равного нулю. Матрицу рассматривать по столбцам. В случае отсутствия положительных или нулевых элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
20. Задан целочисленный двумерный массив A из n строк и m столбцов. Найти минимальное значение положительных элементов, расположенных до первого элемента, равного нулю. Матрицу рассматривать по строкам. В случае отсутствия положительных или нулевых элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
21. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер первого максимального значения среди отрицательных элементов, расположенных после первого элемента, равного T . Матрицу рассматривать по столбцам. В случае отсутствия отрицательных или равных T элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
22. Задан целочисленный двумерный массив A из n строк и m столбцов. Найти номер первого максимального значения среди элементов, меньших A_{11} , и расположенных после первого элемента, кратного трем. Матрицу рассматривать по строкам. В случае отсутствия меньших A_{11} или кратных трем элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
23. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти максимальное значение среди отрицательных элементов, расположенных до первого элемента, равного T . Матрицу рассматривать по столбцам. В случае отсутствия отрицательных или равных T элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
24. Задан целочисленный двумерный массив A из n строк и m столбцов. Найти номер последнего максимального элемента среди элементов, значения которых лежат в диапазоне $[t_1, t_2]$, и расположенных до первого элемента с четным значением. Матрицу рассматривать по строкам. В случае отсутствия элементов в указанном диапазоне значений или четных элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
25. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер последнего максимального значения среди нечетных (по значению) элементов, расположенных до последнего четного (по значению) элемента. Матрицу рассматривать по столбцам. В случае отсутствия нечетных или четных элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
26. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер первого максимального элемента среди положительных элементов, расположенных до последнего отрицательного элемента. Матрицу рассматривать по строкам. В случае отсутствия положительных или отрицательных элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
27. Задан целочисленный двумерный массив A из n строк и m столбцов. Найти максимальное значение положительных элементов, расположенных после первого элемента, кратного пяти. Матрицу рассматривать по столбцам. В случае отсутствия положительных или кратных пяти элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
28. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер последнего минимального значения среди отрицательных элементов, расположенных после первого элемента, равного S . Матрицу рассматривать по строкам. В случае отсутствия отрицательных или равных S элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
29. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер последнего максимального элемента среди положительных элементов, начиная с первого элемента, большего T . Матрицу рассматривать по столбцам. В случае отсутствия положительных или больших T элементов или невозможности поиска вывести соответствующие поясняющие сообщения.

30. Задан целочисленный двумерный массив A из n строк и m столбцов. Найти минимальное значение среди элементов, меньших B , расположенных до первого элемента, большего A_{11} . Матрицу рассматривать по строкам. В случае отсутствия меньших B или больших A_{11} элементов или невозможности поиска вывести соответствующие поясняющие сообщения.
31. Задан целочисленный двумерный массив a из n строк и m столбцов. Найти номер первого максимального элемента среди отрицательных элементов, расположенных до первого элемента, большего T . Матрицу рассматривать по столбцам. В случае отсутствия отрицательных или больших T элементов или невозможности поиска вывести соответствующие поясняющие сообщения.

Список задач по заданию В – «Бинарное дерево».

Для каждого человека в генеалогическом дереве указаны следующие сведения: Фамилия, Имя, Отчество, дата рождения, а также при необходимости дата смерти и место рождения. Для каждой вершины левое поддереве отвечает за отца, правое – за мать. Например, дерево Ивана (указаны для упрощения только имена):



Генеалогическое дерево Ивана.

Отец: Пётр, родителями которого являются Алексей и Елена (дедушка и бабушка Ивана);

Мать: Мария, у которой отец не указан в дереве, а мать Ольга (бабушка Ивана).

- а) Создав соответствующее упорядоченное дерево, вывести информацию обо всех людях в алфавитном (по Ф, И, О) порядке. б) В генеалогическом дереве подсчитать у скольких Иванов отцом был Петр.
- а) Создав соответствующее упорядоченное дерево, вывести информацию обо всех людях от самого старшего до самого младшего по дате рождения. б) В генеалогическом дереве подсчитать у скольких Сергеев бабушкой была Мария.
- а) Создав соответствующее упорядоченное дерево, создать на год (с января по декабрь) план празднования дней рождения. б) По генеалогическому дереву определить, кто дольше всех прожил.

4. а) Создал соответствующее упорядоченное дерево, создать на год (с января по декабрь) план поминок усопших родственников. б) По генеалогическому дереву определить, у скольких родственников имя как у Вас (корневая вершина).
5. а) Создал соответствующее упорядоченное дерево, вывести информацию обо всех людях в алфавитном (по Ф, И, О) порядке. б) В генеалогическом дереве подсчитать количество супружеских пар Иван да Марья.
6. а) Создал соответствующее упорядоченное дерево, вывести информацию обо всех людях от самого старшего до самого младшего по дате рождения. б) В генеалогическом дереве найти всех предков, живших в 19 веке.
7. а) Создал соответствующее упорядоченное дерево, создать на год (с января по декабрь) план празднования дней рождения. б) По генеалогическому дереву определить, кто был мужем Елены (найти все пары, где жена Елена).
8. а) Создал соответствующее упорядоченное дерево, создать на год (с января по декабрь) план поминок усопших родственников. б) По генеалогическому дереву определить, являетесь ли Вы потомком Долгоруких.
9. а) Создал соответствующее упорядоченное дерево, вывести информацию обо всех людях в алфавитном (по Ф, И, О) порядке. б) В генеалогическом дереве подсчитать количество родственников, родившихся в Киеве.
10. а) Создал соответствующее упорядоченное дерево, вывести информацию обо всех людях от самого старшего до самого младшего по дате рождения. б) В генеалогическом дереве найти всех предков, кто родился в Москве. прожил не менее 70 лет.
11. а) Создал соответствующее упорядоченное дерево, создать на год (с января по декабрь) план празднования дней рождения. б) По генеалогическому дереву определить, кто был бабушкой Игоря (найти все сочетания).
12. а) Создал соответствующее упорядоченное дерево, создать на год (с января по декабрь) план поминок усопших родственников. б) По генеалогическому дереву определить, есть ли в Вашем роду кто-нибудь, родившийся во Франции.
13. а) Создал соответствующее упорядоченное дерево, вывести информацию обо всех людях в алфавитном (по Ф, И, О) порядке. б) В генеалогическом дереве подсчитать количество современников войны 1812 года.
14. а) Создал соответствующее упорядоченное дерево, вывести информацию обо всех людях от самого старшего до самого младшего по дате рождения. б) В генеалогическом дереве найти всех предков, кто прожил не менее 70 лет.
15. а) Создал соответствующее упорядоченное дерево, создать на год (с января по декабрь) план празднования дней рождения. б) По генеалогическому дереву определить, в каком году родилась София, внучка Петра (найти все сочетания).
16. а) Создал соответствующее упорядоченное дерево, создать на год (с января по декабрь) план поминок усопших родственников. б) По генеалогическому дереву определить, есть ли в Вашем роду кто-нибудь, кто прожил до 100 лет.
17. а) Создал соответствующее упорядоченное дерево, вывести информацию обо всех людях в алфавитном (по Ф, И, О) порядке. б) В генеалогическом дереве подсчитать количество современников войны 1941-1945 года.

18. а) Создав соответствующее упорядоченное дерево, вывести информацию обо всех людях от самого старшего до самого младшего по дате рождения. б) В генеалогическом дереве найти всех предков, кто родился в августе.
19. а) Создав соответствующее упорядоченное дерево, создать на год (с января по декабрь) план празднования дней рождения. б) По генеалогическому дереву определить, в каком году родилась Елена, мама Дмитрия (найти все сочетания).
20. а) Создав соответствующее упорядоченное дерево, создать на год (с января по декабрь) план поминок усопших родственников. б) По генеалогическому дереву определить, есть ли в Вашем роду кто-нибудь, кто родился в указанный день и месяц (задаются в интерактивном режиме).
21. а) Создав соответствующее упорядоченное дерево, вывести информацию обо всех людях в алфавитном порядке по именам, при совпадении имен – по фамилиям. б) В генеалогическом дереве подсчитать количество родственников, родившихся в указанный временной период (задается в интерактивном режиме).
22. а) Создав соответствующее упорядоченное дерево, вывести информацию обо всех людях от самого младшего до самого старшего по дате рождения. б) В генеалогическом дереве найти всех предков, кто прожил менее определенного количества лет (задается в интерактивном режиме).
23. а) Создав соответствующее упорядоченное дерево, создать на год (с января по декабрь) план празднования дней рождения. б) По генеалогическому дереву определить, в каком году во Владимире родился Ярослав (найти все сочетания).
24. а) Создав соответствующее упорядоченное дерево, создать на год (с января по декабрь) план поминок усопших родственников. б) По генеалогическому дереву определить, есть ли в Вашем роду кто-нибудь, у кого обоих дедушек звали Иванами.
25. а) Создав соответствующее упорядоченное дерево, вывести информацию обо всех людях в алфавитном порядке по именам, при совпадении имен – по отчествам. б) В генеалогическом дереве подсчитать количество родственников, про одного или обоих родителей которого у Вас нет информации.
26. а) Создав соответствующее упорядоченное дерево, вывести информацию обо всех людях от самого младшего до самого старшего по дате рождения. б) В генеалогическом дереве найти всех предков, кто родился в том же городе, что и его жена.
27. а) Создав соответствующее упорядоченное дерево, создать на год (с января по декабрь) план празднования дней рождения. б) По генеалогическому дереву определить, где родился Ярослав, внук Дмитрия (найти все сочетания).
28. а) Создав соответствующее упорядоченное дерево, создать на год (с января по декабрь) план поминок усопших родственников. б) По генеалогическому дереву определить, есть ли в Вашем роду кто-нибудь, у кого обоих бабушек звали одинаковым именем.
29. а) Создав соответствующее упорядоченное дерево, вывести информацию обо всех людях в алфавитном порядке по именам, при совпадении имен – по фамилиям. б) В генеалогическом дереве подсчитать количество родственников указанным именем (задается в интерактивном режиме), и вывести информацию о них.

30. а) Создав соответствующее упорядоченное дерево, создать на год (с января по декабрь) план празднования дней рождения. б) По генеалогическому дереву определить, в каком году в указанном городе (задается в интерактивном режиме) родилась Мария (найти все сочетания).

31=1

32=2

33=3

...