На экзамен по дисциплине «ОА и П» (второй семестр) выносятся вопросы (задачи) по следующим темам:

* Указатели на функции.
* Структуры, статические и динамические массивы структур.
* Объединения.
* Поля бит.
* Списки (однонаправленные, двунаправленные).
* Стек (решение задач с использованием стека, проверка правильности расстановок скобок в арифметическом выражении, обратная польская запись и др.).
* Очередь (однонаправленная, двунаправленная, дек), основные операции с очередью, применение при решении задач.
* Кольцо (однонаправленное, двунаправленное), основные операции с кольцом, приме- нение при решении задач.
* Бинарные деревья, основные операции с деревьями, применение при решении задач.

В задачах по перечисленным темам могут встречаться вопросы по материалу преды- дущего семестра, так как информационно материал в обоих семестрах тесно связан.

Все задачи, содержащиеся в экзаменационных билетах – типовые. Аналогичные зада- чи (задачи с однотипным подходом к решению) рассмотрены: либо на лекциях по курсу, ли- бо на лабораторных работах, либо имеются в учебном пособии данной дисциплины.

Список задач из которого составляются экзаменационные билеты:

1. Для уже существующего дискового файла выполнить сортировку его элементов методом ”через отбор”. (Дополнительных массивов и файлов не использовать). Файл – текстовый. В файле записаны только 1 и 2-х значные числа.
2. Для уже существующего дискового файла выполнить сортировку его элементов методом “вставок”. (Дополнительных массивов и файлов не использовать). Файл – би- нарный.
3. Написать функцию удаления любого элемента из списка имеющего организацию –

двунаправленное кольцо.

1. Для существующего, упорядоченного по возрастанию, бинарного файла ввести с клавиатуры несколько чисел, размещая их в файле без нарушения его упорядоченности. Дополнительных массивов и файлов не использовать.
2. Из командной строки получить имя бинарного файла и границы диапазона чисел, которые должны быть удалены из него. Дополнительных массивов и файлов не исполь- зовать.
3. Файл содержит информацию о группе учащихся (фамилия и 3 оценки). Удалить из файла информацию о всех учащихся с минимальной суммой баллов. Имя файла получить из командной строки. Тип файла выбрать самостоятельно.
4. Написать функцию сортировки списка имеющего организацию – двунаправленное кольцо изменением указателей в нем.
5. Выполнить кодирование текстового файла содержащего слова следующим обра- зом: первую букву поменять с последней, вторую с предпоследней, и так далее. Допол- нительных файлов и массивов не использовать.
6. В бинарном числовом файле выполнить перестановку местами 1 и 2, 3 и 4, и так далее элементов. Дополнительных файлов и массивов не использовать.
7. Написать функцию удаления любого элемента. из однонаправленной очереди. Элемент очереди содержит указатель на символьную строку.
8. Из однонаправленного кольца, элементами которого являются указатели на стек и символьную строку (фамилию) удалить элемент с MAX фамилией, а стек добавить к стеку следующего элемента кольца.
9. Написать функцию нерекурсивную создания бинарного дерева элементами кото- рого являются фамилия (ключ) и год рождения .
10. Написать функцию сортировки списка имеющего организацию – однонаправлен- ное кольцо перемещением содержимого элементов списка. Метод сортировки – ”встав- ками”.
11. Написать функцию добавления в бинарное дерево информации из текстового фай- ла, содержащего символьные строки. В бинарное дерево заносить отдельные слова.
12. Написать рекурсивную функцию создания бинарного дерева элементами которого являются фамилия и год рождения (ключ).
13. Написать функцию сохранения информации из бинарного дерева в бинарный файл (должен быть упорядочен по убыванию). Имя файла получено из командной строки.
14. Написать рекурсивную функцию создания двунаправленного кольца.
15. Написать функцию сохранения информации из упорядоченной по убыванию од- нонаправленной очереди в бинарный файл (должен быть упорядочен по возрастанию). Имя получено из командной строки. Дополнительных массивов и файлов не применять.
16. Организовать программу-библиотечный каталог. Использовать массив структур поля которой – автор, название книги, год издания. Вносимая информация должна быть отсортирована вначале по автору далее по названию.
17. Написать функцию ввода информации в массив организованный по принципу би- нарного дерева (для k-го элемента-предка потомками будут 2\*k-левый сын и 2\*k+1 – правый).
18. Для массива структур, состоящего из полей: указатели на две структуры 1-я - сту- дент и номер группы, 2-я - 4 оценки, используя указатели организовать поиск и вывод на экран информации о студенте с max суммой баллов.
19. Написать функцию сортировки списка имеющего организацию – двунаправленная очередь изменением указателей в ней.
20. Написать функцию сортировки списка имеющего организацию – однонаправлен- ное кольцо изменением указателей в нем.
21. Для уже существующего дискового файла выполнить сортировку его элементов методом ”через отбор”. (Дополнительных массивов и файлов не использовать). Файл – текстовый. В файле записаны только 1 и 2-х значные числа.
22. Для уже существующего дискового файла выполнить сортировку его элементов методом “вставок”. (Дополнительных массивов и файлов не использовать). Файл – би- нарный.
23. Для существующего, упорядоченного по возрастанию, бинарного файла ввести с клавиатуры несколько чисел, размещая их в файле без нарушения его упорядоченности. Дополнительных массивов и файлов не использовать.
24. Написать функцию удаления из текстового файла всех слов, содержащих заданный символ. Дополнительных массивов и файлов не использовать.
25. Имеется бинарный файл, содержащий наименование товара и его количество на складе. Написать функцию удаления из файла всех товаров отсутствующих на складе (т.е. с количеством равным 0). Дополнительных массивов и файлов не использовать.
26. Имеется бинарный файл, содержащий информацию о наименовании товара и его количество на складе. Написать функцию сжатия информации в файле (т.е. для каждого

наименования товара в файле получается его остаток, а остальная информация о нем уда- ляется из файла). Дополнительных массивов и файлов не использовать.

1. Существующее бинарное дерево переписать в однонаправленное кольцо. Содер- жимое узла бинарного дерева – название города. Кольцо должно быть упорядочено по алфавиту (по возрастанию). Сортировок не использовать.
2. Написать функцию рекурсивного прохода по бинарному дереву и вывода его со- держимого по убыванию.
3. Написать функцию удаления из бинарного дерева любого узла.
4. Написать функцию сортировки списка имеющего организацию – однонаправлен- ная очередь изменением указателей в ней. Содержимое элементов очереди – символьные строки (указатели на них).
5. Написать функцию нерекурсивного прохода по бинарному дереву и вывода его содержимого (вначале выводятся узлы-потомки затем узлы-предки)
6. Написать функцию нерекурсивного прохода по бинарному дереву и вывода его содержимого (вначале выводятся узлы-предки затем их узлы-потомки)
7. Написать функцию нерекурсивного прохода по бинарному дереву и вывода его содержимого (вначале выводятся узлы-потомки затем узлы-предки)
8. Разработать функцию в которой вводимые с клавиатуры целые числа удаляются из бинарного файла. Дополнительных массивов и файлов не применять.
9. Написать функцию удаления из текстового файла всех совпадающих символов. Дополнительных массивов и файлов не использовать.
10. Написать функцию прохода по однонаправленному кольцу. Содержимое элемен- тов кольца – целые числа. Элементы, нарушающие упорядоченность по возрастанию ин- формации в кольце удалить.
11. Написать функцию удаления любого элемента из списка имеющего огранизацию двунаправленного кольца.
12. Написать функцию добавления в список, имеющий организацию - однонаправ- ленная очередь, новых элементов содержащих фамилии студентов некоторой учебной группы. Содержимое элементы очереди – фамилия. Очередь упорядочена по алфавиту и вводимые элементы не должны нарушать ее упорядоченности.
13. Для уже существующего текстового файла (файл отсортирован) разработать функ- цию ввода с клавиатуры символов (только русских букв) и добавления их в файл без нарушения его упорядоченности. Сортировки не применять, дополнительных массивов и файлов не использовать.
14. Написать функцию рекурсивного прохода по бинарному дереву и вывода его со- держимого по возрастанию.
15. Написать функцию удаления любого элемента из списка, имеющего организацию однонаправленной очереди.
16. Используя поля битов, найти остаток от деления введенного числа на 2, 4.
17. Имеются 2 файла отсортированных, один по возрастанию, другой по убыванию. Переписать их в третий файл, упорядочивая по возрастанию. Все файлы бинарные. Сор- тировок не применять, дополнительных массивов и файлов не использовать.
18. Имеются 2 файла отсортированных оба по возрастанию. Переписать их в третий файл, упорядочивая по возрастанию. Все файлы бинарные. Сортировок не применять, дополнительных массивов и файлов не использовать.
19. Имеются 2 файла отсортированных оба по возрастанию. Переписать их в третий файл, упорядочивая по убыванию. Все файлы бинарные. Сортировок не применять, до- полнительных массивов и файлов не использовать.
20. Имеются 2 файла отсортированных, один по возрастанию, другой по убыванию. Переписать их в третий файл, упорядочивая по убыванию. Все файлы бинарные. Сорти- ровок не применять, дополнительных массивов и файлов не использовать.
21. Имеются 2 файла отсортированных один по возрастанию, другой по убыванию. Переписать их в третий файл, упорядочивая по возрастанию. Все файлы текстовые. Сор- тировок не применять, дополнительных массивов и файлов не использовать.
22. Имеются 2 файла отсортированных оба по возрастанию. Переписать их в третий файл, упорядочивая по возрастанию. Все файлы текстовые. Сортировок не применять, дополнительных массивов и файлов не использовать.
23. Имеются 2 файла отсортированных оба по возрастанию. Переписать их в третий файл, упорядочивая по убыванию. Все файлы текстовые. Сортировок не применять, до- полнительных массивов и файлов не использовать.
24. Имеются 2 файла отсортированных один по возрастанию, другой по убыванию. Переписать их в третий файл, упорядочивая по убыванию. Все файлы текстовые. Сорти- ровок не применять, дополнительных массивов и файлов не использовать.
25. Используя поля битов вывести битовое представление ASCII-кода символа алфа- вита на экран.
26. Выполнить реверсивный переворот числовой информации в текстовом файле (2- значные числа) следующим образом: первое число поменять с последним, второе с пред- последним, и так далее. Рекурсию не применять. Дополнительных массивов и файлов не использовать.
27. Написать программу удаления из бинарного файла чисел содержащихся в тексто- вом файле. Имена файлов получить из командной строки. В файлах содержится только цифровая информация.
28. Написать функцию добавления в отсортированный бинарный файл целых чисел без нарушения упорядоченности в нем. Сортировок не использовать, дополнительных массивов и файлов не использовать.
29. Написать программу удаления из текстового файла (числа от 1 до 999) чисел со- держащихся в бинарном файле. Имена файлов получить из командной строки. Дополни- тельных массивов и файлов не использовать.
30. Написать функцию удаления любого элемента из списка имеющего организацию однонаправленного кольца.
31. Используя структуры, полями которой являются: имя фигуры, объединения (со- держащие параметры геометрических фигур – прямоугольник, круг) и переменная – площадь, написать программу вначале вычисления площадей 100 фигур, затем вывода (наименование -площадь).
32. Для уже существующего бинарного файла длинных целых чисел выполнить сор- тировку его элементов методом ”через отбор”. Дополнительных массивов и файлов не использовать.
33. Написать функцию нерекурсивного прохода по бинарному дереву и вывода его содержимого (вначале выводятся узлы-предки затем их узлы-потомки)
34. Написать функцию удаления из линейного двунаправленного списка последнее вхождения заданного элемента.
35. Написать функцию удаления первого вхождения заданного элемента из линейного однонаправленного списка.
36. Написать функцию удаления из кольцевого двунаправленного списка последнего вхождения заданного элемента.
37. Написать функцию удаления первого вхождения заданного элемента из кольцево- го однонаправленного списка.
38. Написать функцию перезаписи линейного однонаправленного списка в обратном порядке.
39. Написать функцию перезаписи кольцевого однонаправленного списка в обратном порядке.
40. Написать функцию объединения двух линейных двунаправленных списков без дублирования элементов.
41. Написать функцию объединения двух кольцевых двунаправленных списков без дублирования элементов.
42. Написать функцию замены местами элементов с нечетными индексами с элемен- тами с четными индексами в кольцевом двунаправленном списке.
43. Написать функцию замены местами элементов с нечетными индексами с элемен- тами с четными индексами в линейном двунаправленном списке.
44. Разработать функцию подсчета числа узлов заданного бинарного дерева.
45. Разработать программу проверки находится ли элемент с ключом ’A’ в поддереве с корнем в вершине в ключом ’B’.
46. Разработать функцию подсчета числа ”листьев” заданного бинарного дерева.
47. Разработать функцию подсчета в заданном бинарном дереве узлов имеющих два наследника.
48. Разработать функцию глубины заданного бинарного дерева.
49. Разработать функцию проверки является ли заданное бинарное дерево закончен- ным.
50. Разработать функцию проверки является ли заданное бинарное дерево полным.
51. Разработать функцию проверки является ли заданное бинарное дерево вырожден- ным.
52. Дан файл f, компоненты которого являются действительными числами. Найти про- изведение компонент файла.
53. Дан файл f, компоненты которого являются целыми числами. Никакая из компо- нент файла не равна нулю. Файл f содержит столько же отрицательных чисел, сколько и положительных. Используя вспомогательный файл h, переписать компоненты файла f в файл g так, чтобы в файле g сначала шли положительные, потом отрицательные числа
54. Дан файл f, компоненты которого являются целыми числами. Получить в файле g все компоненты файла f, являющиеся точными квадратами.
55. Дан файл f, компоненты которого являются действительными числами. Найти сум- му наибольшего и наименьшего из значений компонент.
56. Дан файл, содержащий различные даты. Каждая дата – это число, месяц и год. Найти год с наименьшим номером.
57. Дан файл f, компоненты которого являются действительными числами. Найти мо- дуль суммы и квадрат произведения компонент файла.
58. Дан файл f, компоненты которого являются действительными числами. Найти раз- ность первой и последней компонент файла.
59. Дан файл f, компоненты которого являются целыми числами. Найти количество чётных чисел среди компонент.
60. Дан символьный файл f. Получить копию файла в файле g.
61. Даны символьные файлы f1 и f2. Переписать с сохранением порядка следования компоненты файла f1 в файл f2. Использовать вспомогательный файл h.
62. Дан файл f, компоненты которого являются действительными числами. Найти сум- му компонент файла.
63. Дан символьный файл f. В файле не менее двух компонент. Определить, являются ли два первых символа файла цифрами. Если да, то установить, является ли число, обра- зованное этими цифрами чётным.
64. Дан файл f, компоненты которого являются целыми числами. Получить в файле g все компоненты файла f являющиеся чётными числами.
65. Дан файл f, компоненты которого являются действительными числами. Найти наибольшее из значений модулей компонент с нечётными номерами.
66. Дан файл f, компоненты которого являются действительными числами. Найти по- следнюю компоненту файла
67. Дан файл, содержащий различные даты. Каждая дата - это число, месяц и год. Найти все весенние даты.
68. Дан файл f, компоненты которого являются целыми числами. Получить в файле g все компоненты файла f, делящиеся на 3 и не делящиеся на 7.
69. Дан файл f, компоненты которого являются действительными числами. Найти наименьшее из значений компонент с чётными номерами.
70. Записать в файл g все чётные числа файла f, а в файл h все нечётные. Порядок сле- дования чисел сохраняется.
71. Дан символьный файл f .Записать в файл g компоненты файла f в обратном поряд- ке.
72. Дан файл, содержащий различные даты. Каждая дата - это число, месяц и год. Найти самую позднюю дату
73. Даны символьные файлы f и g. Записать в файл h сначала компоненты файла f, за- тем компоненты файла g с сохранением порядка.
74. Дан файл f, компоненты которого являются целыми числами. Никакая из компо- нент файла не равна нулю. Файл f содержит столько же отрицательных чисел, сколько и положительных. Используя вспомогательный файл h, переписать компоненты файла f в файл g так, чтобы в файле g не было двух соседних чисел с одним знаком.
75. Дан файл f, компоненты которого являются целыми числами. Найти количество квадратов нечётных чисел среди компонент.
76. Дан файл f, компоненты которого являются действительными числами. Найти сум- му квадратов компонент файла.
77. Дан файл f, компоненты которого являются действительными числами. Найти раз- ность кубов компонент файла.
78. Дан файл f, компоненты которого являются целыми числами. Найти количество удвоенных нечётных чисел среди компонент.
79. Дан файл f, компоненты которого являются целыми числами. Никакая из компо- нент файла не равна нулю. Файл f содержит столько же отрицательных чисел, сколько и положительных. Используя вспомогательный файл h, переписать компоненты файла f в файл g так, чтобы в файле g числа шли в следующем порядке: два положительных два отрицательных, два положительных, два отрицательных и т.д. (предполагается, что число компонент в файле f делится на 4).
80. Дан файл, содержащий различные даты. Каждая дата - это число, месяц и год. Найти самую раннюю дату.
81. Дан файл f, компоненты которого являются целыми числами. Никакая из компо- нент файла не равна нулю. Файл f содержит столько же отрицательных чисел, сколько и положительных. Используя вспомогательный файл h, переписать компоненты файла f в файл g так, чтобы в файле g сначала шли нечетные потом четные числа
82. В первом файле хранится k матриц размерности m x n, во втором - l матриц раз- мерности m x n. Те матрицы из первого файла, у которых 0 a00  , перенести в конец вто- рого файла. Вывести на экран содержимое первого и второго файлов.
83. В первом файле хранится k матриц размерности m x n, во втором l матриц размер- ности m x n. Убрать из файла, в котором больше матриц, лишние матрицы в третий файл. Вывести на экран содержимое первого файла; второго файла; третьего файла.
84. Файл состоит из k компонент структуры, где каждая компонента содержит две мат- рицы: первая размерности m x n, вторая размерности m x l. Получить k произведений со- ответствующих матриц и записать их во второй файл. Вывести на экран содержимое пер- вого и второго файлов.
85. В первом файле хранится k матриц размерности m x n, во втором l матриц размер- ности m x n. Добавить во второй файл те матрицы из первого, которых нет во втором. Вывести на экран содержимое первого и второго файлов.
86. В первом файле хранится k матриц из n строк и n+1 столбцов каждая (последний столбец - столбец свободных членов). Во втором файле хранится k векторов - результа- тов решений соответствующих систем ЛАУ с матрицами из первого файла. Вывести на экран покомпонентно исходную систему уравнений и результат, проверив его предвари- тельно; добавить в файлы новые данные; удалить ненужную информацию.
87. В файле хранится k матриц размерности m× n. Для каждой матрицы из файла вы- числить сумму её положительных четных элементов. Все матрицы с четными суммами записать в другой файл, заменив их в исходном файле единичными матрицами. Вывести на экран содержимое первого и второго файлов.
88. В первом файле хранится k матриц размерности m× n, во втором - l матриц размер- ности m× n. Поменять местами все нечетные (по порядковому номеру в файле) матрицы из первого и второго файлов (до конца меньшего из файлов). Вывести на экран содержи- мое первого и второго файлов.
89. В первом файле хранится k квадратных матриц порядка n, во втором – l квадратных матриц .Если k≠1, то в файл с меньшим числом матриц добавить в конец файла недоста- ющее количество единичных матриц. Вывести на экран содержимое первого и второго файлов.
90. В файле хранится k матриц размерности n× n. Для каждой матрицы из файла вы- числить сумму её диагональных элементов. Все матрицы с нечетными суммами записать в другой файл, заменив их в исходном файле транспонированными матрицами. Вывести на экран содержимое первого и второго файлов.
91. В первом файле хранится k квадратных матриц . Записать в другой файл из исход- ного файла все симметрические матрицы ( T A  A ), в третий файл – остальные. Вывести на экран содержимое первого, второго и третьего файлов.
92. В первом файле хранится k матриц порядка m × n, во втором - l матриц. Поменять местами все нечетные (1, 3, 5,… по порядковому номеру в файле) матрицы из первого файла с четными матрицами (0, 2, 4, …) второго файла (до конца меньшего из файлов). Оставшиеся в большем файле матрицы переписать в третий файл. Вывести на экран со- держимое первого и второго файлов.
93. В первом файле хранится k матриц размерности m × n, во втором - l матриц раз- мерности m × n. Те матрицы из первого файла, сумма элементов главной диагонали кото- рых равна 5, перенести в конец второго файла. Вывести на экран содержимое первого и второго файлов.
94. В первом файле хранится k матриц размерности m x n, во втором l матриц размер- ности m× n. Убрать из файла, в котором меньше матриц, лишние матрицы в третий файл. Вывести на экран содержимое первого файла; второго файла; третьего файла.
95. Файл состоит из k компонент структуры, где каждая компонента содержит две мат- рицы: первая размерности m × n, вторая размерности m × l. Получить k сумму соответ- ствующих матриц и записать их во второй файл. Вывести на экран содержимое первого и второго файлов.
96. В первом файле хранится k матриц размерности m × n, во втором l матриц размер- ности m × n. Добавить во второй файл те матрицы, определитель которых равен 5. Выве- сти на экран содержимое первого и второго файлов.
97. В первом файле хранится k матриц из n строк и n+1 столбцов каждая (последний столбец - столбец свободных членов). Во втором файле хранится k векторов. Вывести на экран скалярное произведения k матриц на соответствующие вектора, результат, прове- рив его предварительно, добавить в файлы новые данные, удалить ненужную информа- цию.
98. В файле хранится k матриц размерности m× n. Для каждой матрицы из файла вы- числить сумму её отрицательных нечетных элементов. Все матрицы с нечетными сумма- ми записать в другой файл, заменив их в исходном файле единичными матрицами. Выве- сти на экран содержимое первого и второго файлов.
99. В первом файле хранится k матриц размерности m × n, во втором - l матриц раз- мерности m x n. Поменять местами все четные (по порядковому номеру в файле) матри- цы из первого и второго файлов (до конца меньшего из файлов). Вывести на экран со- держимое первого и второго файлов.
100. В первом файле хранится k квадратных матриц порядка n , во втором – l квадрат- ных матриц. Если k≠1 , то в файл с большим числом матриц добавить в начало файла недостающее количество единичных матриц. Вывести на экран содержимое первого и второго файлов.
101. В файле хранится k матриц размерности n × n. Для каждой матрицы из файла вы- числить разность её диагональных элементов. Все матрицы с четными разностями запи- сать в другой файл, заменив их в исходном файле обратными матрицами. Вывести на экран содержимое первого и второго файлов.
102. В первом файле хранится k квадратных матриц. Записать в другой файл из исход- ного файла все обратные матрицы ( 1 A  A ), в третий файл – остальные. Вывести на экран содержимое первого, второго и третьего файлов.
103. В первом файле хранится k матриц размерности m × n, во втором – k матриц раз- мерности m × 1. Получить k разность соответствующих матриц из первого и второго файлов и записать их в третий файл в виде компонент структуры, где каждая компонента содержит три матрицы: первая размерности m × n из первого файла; вторая размерности m × 1 из второго файла; третья, матрица размерности n × 1, результат произведения. Вы- вести на экран содержимое первого и второго файлов.
104. В первом файле хранится k матриц порядка m × n, во втором - l матриц Поменять местами все четные (2, 4, 6,… по порядковому номеру в файле) матрицы из первого фай- ла с нечетными матрицами (1, 3, 5, …) второго файла (до конца меньшего из файлов). Оставшиеся в меньшем файле матрицы переписать в третий файл. Вывести на экран со- держимое первого и второго файлов.
105. В первом файле хранится k матриц размерности m × n, во втором - l матриц раз- мерности m × n. Те матрицы из первого файла, у которых сумма первой строки больше 5 перенести в конец второго файла. Вывести на экран содержимое первого и второго фай- лов.
106. В первом файле хранится k матриц размерности m × n, во втором l матриц размер- ности m × n. Записать в третий файл перемножение матриц k. Вывести на экран содер- жимое первого, второго и третьего файлов.
107. Файл состоит из k компонент структуры, где каждая компонента содержит две мат- рицы: первая размерности m × n, вторая размерности m × l. Записать в третий файл все матрицы, в которых первый столбец равен m× l. Вывести на экран содержимое первого, второго и третьего файлов
108. Ввести две даты и определить количество лет между ними
109. Ввести две даты и вывести сообщение с информацией, какая из двух введенных дат более поздняя.
110. Написать программу любителям предсказаний, позволяющую для введенной даты определить, принадлежит ли она к високосному году, если это не так, то выдать инфор- мацию о количестве лет до ближайшего високосного года.
111. Удалите из дерева все равные между собой элементы. В программе используйте подпрограммы.
112. Удалите из дерева все повторяющиеся элементы. В программе используйте под- программы.
113. Постройте два дерева. Проверьте, является ли одно из них поддеревом другого. Ес- ли "да", то удалите это поддерево. В программе используйте подпрограммы.
114. Постройте два дерева. Проверьте, является ли одно из них поддеревом другого. Ес- ли "нет", то включите это поддерево. В программе используйте подпрограммы.
115. Используя очередь или стек, вычислите среднее арифметическое всех элементов непустого дерева Т и удалите все элементы меньшие этого числа. В программе исполь- зуйте подпрограммы.
116. Используя очередь или стек, поменяйте местами максимальный и минимальный элементы непустого дерева Т, все элементы которого различны. В программе используй- те подпрограммы.
117. Используя очередь или стек, напечатайте все элементы дерева Т по уровням: сна- чала – из корня дерева, затем (слева направо) – из вершин, дочерних по отношению к корню, затем (также слева направо) – из вершин, дочерних по отношению к этим верши- нам, и т.д. В программе используйте подпрограммы.
118. Используя очередь или стек, найдите в непустом дереве Т длину (число ветвей) пу- ти от корня до ближайшей вершины с элементом Е. Если такого элемента не обнаружено, то выдайте на экран соответствующее сообщение. В программе используйте подпро- граммы.
119. Используя очередь или стек, подсчитайте число вершин на n-ом уровне непустого дерева Т (корень считайте вершиной 0-го уровня). В программе используйте подпро- граммы.
120. Создайте программой числовое двоичное дерево. Опишите рекурсивную логиче- скую функцию, проверяющую наличие заданного числа в сформированном дереве. В программе используйте подпрограммы.
121. Создайте программой числовое двоичное дерево. Опишите рекурсивную числовую функцию, подсчитывающую сумму элементов дерева. В программе используйте подпро- граммы.
122. Создайте программой числовое двоичное дерево. Опишите функцию, которая находит наибольший элемент непустого дерева. В программе используйте подпрограм- мы.
123. Напишите программу, содержащую функцию, которая каждый отрицательный элемент дерева заменяет на положительный, а положительный превращает в ноль.
124. Напишите программу, содержащую функцию, которая каждый элемент дерева воз- водит в квадрат.
125. Создайте программой символьное двоичное дерево. Опишите функцию, возвраща- ющую строку, сформированную на базе этих символов. В программе используйте под- программы.
126. Создайте программой символьное двоичное дерево. Опишите логическую функ- цию, проверяющую, есть ли в непустом дереве хотя бы два одинаковых символа. В про- грамме используйте подпрограммы.
127. Создайте строковое двоичное дерево. Опишите функцию, возвращающую строку, сформированную на базе символов, встречающихся в каждой строке дерева. В программе используйте подпрограммы.
128. Создайте двоичное дерево записей. Проверьте выбранное поле записей на равен- ство. В программе используйте подпрограммы.
129. Создайте программой два числовых двоичных дерева. Опишите рекурсивно и нерекурсивно логическую функцию, входными параметрами которой являются два дере- ва, проверяющую на равенство эти деревья. В программе используйте подпрограммы.
130. Напишите программу, содержащую функцию или функцию, которая присваивает параметру Е элемент из самого левого листа непустого дерева (лист – вершина, из кото- рой не выходит ни одной ветви), используя очередь или стек. В программе используйте подпрограммы.
131. Напишите программу, содержащую функцию или функцию, которая находит в не- пустом дереве длины (число ветвей) путей от корня до всех вершин, используя очередь или стек. В программе используйте подпрограммы.
132. Напишите программу, содержащую функцию или функцию, которая подсчитывает число вершин на каждом уровне непустого дерева (корень считать вершиной 0-го уров- ня). В программе используйте подпрограммы.
133. Напишите программу, содержащую функцию или функцию, которая определяет максимальную глубину непустого дерева Т, т.е. число ветвей в самом длинном из путей от еорня дерева до листьев. В программе используйте подпрограммы.
134. Напишите программу, содержащую функцию, которая строит Т1 – копию дерева Т. В программе используйте подпрограммы.
135. Напишите программу, содержащую функцию Create(T,n), где n – положительное целое число, которая строит Т – дерево, изображенное на рисунке. В программе исполь- зуйте подпрограммы.
136. Объедините два дерева в одно идеально сбалансированное. В программе исполь- зуйте подпрограммы.
137. Создать стек для целых чисел. Максимальный размер стека вводится с экрана. Со- здать функции для ввода и вывода элементов стека. Добавить 6 элементов. Удалить и вывести на экран 2 элемента.
138. Создать стек для целых (положительных и отрицательных) чисел. Максимальный размер стека вводится с экрана. Создать функции для ввода и вывода элементов стека. Ввести с экрана 6 элементов. При вводе чисел в стек попадают только отрицатель- ные элементы. Вывести все элементы стека.
139. Создать стек для целых чисел. Максимальный размер стека вводится с экрана. Со- здать функции для ввода, вывода и определения размера стека. Ввести с экрана 6 элемен- тов. Удалить 2 элемента. Вывести размер стека.
140. Создать стек для целых (положительных и отрицательных) чисел. Максимальный размер стека вводится с экрана. Создать функции для ввода, вывода и определения раз- мера стека. Вводить с экрана числа, причем в стек должны добавляться поочередно по- ложительные и отрицательные числа.
141. Создать стек для символов. Максимальный размер стека вводится с экрана. Создать функции для ввода и вывода элементов стека. Ввести эталонный символ. Вводить сим- волы с экрана в стек до встречи эталонного. Вывести все элементы стека.
142. Создать стек для символов. Максимальный размер стека вводится с экрана. Создать функции для ввода и вывода элементов стека. Добавить символы с экрана в стек. После добавления 5-го символа перед добавлением удалять элемент из стека.
143. Создать стек для символов. Максимальный размер стека вводится с экрана. Создать функции для ввода, вывода и определения размера стека. Ввести эталонный символ. Вво- дить символы с экрана в стек до встречи эталонного. Вывести размер стека.
144. Создать стек для символов. Максимальный размер стека вводится с экрана. Создать функции для ввода и вывода элементов стека. Добавлять символы с экрана в стек. В слу- чае совпадения вводимого символа с вершиной стека вытолкнуть его и распечатать ее.
145. Создать стек для символов. Максимальный размер стека вводится с экрана. Создать функции для ввода, вывода и определения размера стека. Вводить символы с экрана в стек. В случае совпадения вводимого символа с вершиной стека вывести размер стека.
146. Создать два стека для символов. Максимальный размер стеков вводится с экрана. Создать функции для ввода и вывода элементов стека. Вводить символы с экрана в пер- вый стек. В случае совпадения вводимого символа с вершиной стека вводить во второй стек.
147. Создать два стека для символов. Максимальный размер стеков вводится с экрана. Создать функции для ввода и вывода элементов стека. Вводить символы с экрана в стеки поочередно.
148. Создать стек для символов и стек для чисел. Максимальный размер стеков вводит- ся с экрана. Создать функции для ввода и вывода элементов стека. Вводить символы с экрана. Символ попадает в первый стек, а его численное представление – во второй.
149. Создать два стека для символов. Максимальный размер стеков вводится с экрана. Создать функции для ввода и вывода элементов стека. Вводить символы с экрана. Про- писные буквы попадают в первый стек, строчные – во второй, остальные символы про- пускаются.
150. Создать два стека для символов. Максимальный размер стеков вводится с экрана. Создать функции для ввода и вывода элементов стека. Вводить символы с экрана. Про- писные буквы преобразуются в строчные и попадают в первый стек, строчные преобра- зуются в прописные и попадают во второй, остальные символы пропускаются.
151. Создать стек для целых чисел. Максимальный размер стека вводится с экрана. Со- здать функции для ввода и вывода элементов стека. Вводить символы с экрана. Числовое представление символа попадает в стек.
152. Создать стек для целых чисел. Максимальный размер стека вводится с экрана. Со- здать функции для ввода и вывода элементов стека. Добавить 6 элементов. Удалить и вывести на экран 2 элемента. Задачу решить с использованием механизма указателей.
153. Создать стек для целых (положительных и отрицательных) чисел. Максимальный размер стека вводится с экрана. Создать функции для ввода и вывода элементов стека. Ввести 6 элементов. При вводе чисел в стек попадают только отрицательные элементы. Вывести все элементы стека. Задачу решить с использованием механизма указателей.
154. Создать стек для целых чисел. Максимальный размер стека вводится с экрана. Со- здать функции для ввода, вывода и определения размера стека. Ввести с экрана 6 элемен- тов. Удалить 2 элемента. Вывести размер стека. Задачу решить с использованием меха- низма указателей.
155. Создать стек для целых (положительных и отрицательных) чисел. Максимальный размер стека вводится с экрана. Создать функции для ввода, вывода и определения раз- мера стека. Вводить с экрана числа, причем в стек должны добавляться поочередно по- ложительные и отрицательные числа. Задачу решить с использованием механизма указа- телей.
156. Создать стек для символов. Максимальный размер стека вводится с экрана. Создать функции для ввода и вывода элементов стека. Ввести эталонный символ. Вводить сим- волы с экрана в стек до встречи эталонного. Вывести все элементы стека. Задачу решить с использованием механизма указателей.
157. Создать стек для символов. Максимальный размер стека вводится с экрана. Создать функции для ввода и вывода элементов стека. Добавить символы с экрана в стек. После

добавления 5-го символа перед добавлением следуюшего удалять элемент из стека. За- дачу решить с использованием механизма указателей.

1. Создать стек для символов. Максимальный размер стека вводится с экрана. Создать функции для ввода, вывода и определения размера стека. Ввести эталонный символ. Вво- дить символы с экрана в стек до встречи эталонного. Вывести размер стека. Задачу ре- шить с использованием механизма указателей.
2. Создать стек для символов. Максимальный размер стека вводится с экрана. Создать функции для ввода и вывода элементов стека. Добавлять символы с экрана в стек. В слу- чае совпадения вводимого символа с вершиной стека вытолкнуть его и распечатать ее. Задачу решить с использованием механизма указателей.
3. Создать стек для символов. Максимальный размер стека вводится с экрана. Создать функции для ввода, вывода и определения размера стека. Вводить символы с экрана в стек. В случае совпадения вводимого символа с вершиной стека вывести размер стека. Задачу решить с использованием механизма указателей.
4. Создать два стека для символов. Максимальный размер стеков вводится с экрана. Создать функции для ввода и вывода элементов стека. Вводить символы с экрана в пер- вый стек. В случае совпадения вводимого символа с вершиной стека вводить во второй стек. Задачу решить с использованием механизма указателей.
5. Создать два стека для символов. Максимальный размер стеков вводится с экрана. Создать функции для ввода и вывода элементов стека. Вводить символы с экрана в стеки поочередно. Задачу решить с использованием механизма указателей.
6. Создать стек для символов и стек для чисел. Максимальный размер стеков вводит- ся с экрана. Создать функции для ввода и вывода элементов стека. Вводить символы с экрана. Символ попадает в первый стек, а его численное представление – во второй. За- дачу решить с использованием механизма указателей.
7. Создать два стека для символов. Максимальный размер стеков вводится с экрана. Создать функции для ввода и вывода элементов стека. Вводить символы с экрана. Про- писные буквы попадают в первый стек, строчные – во второй, остальные символы про- пускаются. Задачу решить с использованием механизма указателей.
8. Создать два стека для символов. Максимальный размер стеков вводится с экрана. Создать функции для ввода и вывода элементов стека. Вводить символы с экрана. Про- писные буквы преобразуются в строчные и попадают в первый стек, строчные преобра- зуются в прописные и попадают во второй, остальные символы пропускаются. Задачу решить с использованием механизма указателей.
9. Создать стек для целых чисел. Максимальный размер стека вводится с экрана. Со- здать функции для ввода и вывода элементов стека. Вводить символы с экрана. Числовое представление символа попадает в стек. Задачу решить с использованием механизма ука- зателей.
10. Создать очередь для целых (положительных и отрицательных) чисел. Максималь- ный размер очереди вводится с экрана. Создать функции для ввода и вывода элементов очереди. Ввести 6 элементов (положительных и отрицательных). Вывести 2 первых от- рицательных элемента очереди.
11. Создать очередь для массива целых (положительных и отрицательных) чисел. Мак- симальный размер очереди вводится с экрана. Создать функции для ввода и вывода эле- ментов очереди. Ввести 6 элементов. При вводе чисел в очередь попадают только отри- цательные элементы. Вывести все элементы очереди.
12. Создать очередь для целых (положительных и отрицательных) чисел. Максималь- ный размер очереди вводится с экрана. Создать функции для ввода, вывода и определе- ния размера очереди. Ввести 6 элементов. Вывести 2 первых элемента очереди. Вывести размер оставшейся очереди.
13. Создать очередь для целых (положительных и отрицательных) чисел. Максималь- ный размер очереди вводится с экрана. Создать функции для ввода, вывода и определе- ния размера очереди. Ввести 6 элементов. Вывести элементы очереди до первого отрица- тельного (включительно). Вывести размер оставшейся очереди.
14. Создать очередь для символов. Максимальный размер очереди вводится с экрана. Создать функции для ввода и вывода элементов очереди. Ввести эталонный символ. Вводить символы с экрана в очередь до встречи эталонного. Вывести все элементы оче- реди.
15. Создать очередь для символов. Максимальный размер очереди вводится с экрана. Создать функции для ввода и вывода элементов очереди. Вводить символы с экрана в очередь. После введения 3-го символа в ответ на каждый вводимый выводить крайний левый символ.
16. Создать очередь для символов. Максимальный размер очереди вводится с экрана. Создать функции для ввода, вывода и определения размера очереди. Ввести эталонный символ. Вводить символы с экрана в очередь до встречи эталонного. Вывести размер очереди.
17. Создать очередь для символов. Максимальный размер очереди вводится с экрана. Создать функции для ввода и вывода элементов очереди. Вводить символы с экрана в очередь. В случае совпадения вводимого символа с последним элементом очереди выво- дить первый элемент очереди.
18. Создать очередь для символов. Максимальный размер очереди вводится с экрана. Создать функции для ввода, вывода и определения размера очереди. Вводить символы с экрана в очередь. В случае совпадения вводимого символа с последним элементом оче- реди выводить размер очереди.
19. Создать очередь для символов. Максимальный размер очереди вводится с экрана. Создать функции для ввода и вывода элементов очереди. Добавлять символы с экрана в очередь. В случае совпадения вводимого символа с последним элементом очереди уда- лять и выводить на экран все элементы очереди.
20. Создать очередь для символов. Максимальный размер очереди вводится с экрана. Создать функции для ввода и вывода элементов очереди. Вводить символы с экрана. В случае совпадения вводимого символа с последним элементом очереди в очередь его не добавлять, а удалять первый элемент.
21. Создать очередь для символов. Максимальный размер очереди вводится с экрана. Создать функции для ввода и вывода элементов очереди. Вводить символы с экрана. В случае совпадения вводимого символа с последним элементом очереди выводить размер очереди.
22. Создать очередь для целых чисел. Максимальный размер очереди вводится с экра- на. Создать функции для ввода и вывода элементов очереди. Ввести в очередь числа с экрана. После этого перейти в режим, при котором при каждом вводе числа из очереди удаляется первый элемент, и если он совпадает с введенным числом, то он добавляется в очередь.
23. Создать очередь для целых чисел. Максимальный размер очереди вводится с экра- на. Создать функции для ввода и вывода элементов очереди. Ввести в очередь числа с экрана. После этого перейти в режим, при котором при каждом вводе числа выводится первый элемент очереди, и если он не совпадает с введенным числом, то оно заносится в очередь.
24. Создать очередь для целых чисел. Максимальный размер очереди вводится с экра- на. Создать функции для ввода и вывода элементов очереди. Ввести в очередь числа с экрана. После этого перейти в режим ввода, при котором перед добавлением элемента происходит удаление одного элемента.
25. Создать дек для целых (положительных и отрицательных) чисел. Максимальный размер дека вводится с экрана. Создать функции для ввода и вывода элементов дека. Ввести 6 элементов (положительных и отрицательных). Вывести 2 первых правых отри- цательных элемента дека.
26. Создать дек для массива целых (положительных и отрицательных) чисел. Макси- мальный размер дека вводится с экрана. Создать функции для ввода и вывода элементов дека. Ввести 3 элемента справа и 3 слева. При вводе чисел в дек попадают только отрица- тельные элементы. Вывести все элементы дека.
27. Создать дек для целых (положительных и отрицательных) чисел. Максимальный размер дека вводится с экрана. Создать функции для ввода, вывода и определения разме- ра дека. Ввести 3 элемента справа и 3 слева. Вывести по одному элементу справа и слева. Вывести размер оставшегося дека.
28. Создать дек для целых (положительных и отрицательных) чисел. Максимальный размер дека вводится с экрана. Создать функции для ввода, вывода и определения разме- ра дека. Ввести 3 элемента справа и 3 слева. Вывести элементы дека справа до первого отрицательного (включительно). Вывести размер оставшегося дека.
29. Создать дек для символов. Максимальный размер дека вводится с экрана. Создать функции для ввода и вывода элементов дека. Ввести эталонный символ. Вводить симво- лы в дек поочередно справа и слева до встречи эталонного. Вывести все элементы дека.
30. Создать дек для символов. Максимальный размер дека вводится с экрана. Создать функции для ввода и вывода элементов дека. Вводить элементы в дек поочередно справа и слева. При этом в ответ на добавление элемента с противоположной стороны дека один элемент удаляется.
31. Создать дек для символов. Максимальный размер дека вводится с экрана. Создать функции для ввода, вывода и определения размера дека. Ввести эталонный символ. Вво- дить символы в дек поочередно справа и слева до встречи эталонного. Вывести размер дека.
32. Создать дек для символов. Максимальный размер дека вводится с экрана. Создать функции для ввода и вывода элементов дека. Добавлять символы в дек поочередно спра- ва и слева. В случае совпадения добавляемого символа с элементом на другом конце дека выводить его на экран.
33. Создать дек для символов. Максимальный размер дека вводится с экрана. Создать функции для ввода, вывода и определения размера дека. Добавлять символы в дек пооче- редно справа и слева. В случае совпадения добавляемого символа с элементом на другом конце дека выводить размер дека.
34. Создать дек для символов. Максимальный размер дека вводится с экрана. Создать функции для ввода и вывода элементов дека. Добавлять символы в дек поочередно спра- ва и слева. В случае совпадения добавляемого символа с концом дека, вывести на экран все элементы со стороны совпавшего.
35. Создать дек для символов. Максимальный размер дека вводится с экрана. Создать функции для ввода и вывода элементов дека. Добавлять символы в дек поочередно спра- ва и слева. В случае совпадения введенного символа с элементом соответствующего кон- ца дека его не добавлять.
36. Создать дек для символов. Максимальный размер дека вводится с экрана. Создать функции для ввода, вывода и определения размера дека. Ввести в дек символы с экрана. Вводить символы с экрана. В случае совпадения вводимого символа с одним из концов дека выводить его размер.
37. Создать дек для целых чисел. Максимальный размер дека вводится с экрана. Со- здать функции для ввода и вывода элементов дека. При каждом вводе числа слева удаля- ется элемент, и если он не совпадает с введенным числом, то введенное число добавляет-

ся справа.

1. Создать дек для целых чисел. Максимальный размер дека вводится с экрана. Со- здать функции для ввода и вывода элементов дека. Ввести в дек числа с экрана. После этого перейти в режим, при котором слева удаляется элемент, и если он совпадает с вве- денным числом, то введенное число добавляется справа, а иначе слева.
2. Создать дек для плавающих чисел. Максимальный размер дека вводится с экрана. Создать функции для ввода и вывода элементов дека. Ввести в дек числа с экрана. После этого перейти в режим ввода, при котором перед занесением элемента происходит удале- ние левого элемента.
3. Введите строку. Проанализировав в программе содержимое строки, выберите из нее числа и занесите в очередь. Выведите содержимое очереди на экран и посчитайте сумму этих чисел. Решение в программе оформляйте через подпрограммы.
4. Введите строку. Проанализировав в программе содержимое строки, выберите из нее имена и занесите в очередь. Выведите содержимое очереди на экран и посчитайте ко- личество элементов образованной очереди. Решение в программе оформляйте через под- программы.
5. Проверьте на равенство две очереди. Решение в программе оформляйте через под- программы.
6. Найдите среди трех очередей две одинаковые. Решение в программе оформляйте через подпрограммы.
7. Организовать три очереди с одинаковым количеством элементов, содержащие со- ответствено имена, отчества и фамилии людей. Составьте очередь из элементов, содер- жащих наиболее полную информацию о людях, воспользовавшись уже созданными оче- редями и запросив какую-то дополнительную информацию. Решение в программе оформляйте через подпрограммы.
8. Создайте строку. Организовывая очереди по N элементов, создайте строки по N символов в каждой. Решение в программе оформляйте через подпрограммы.
9. Создайте строку. Проанализировав в программе содержимое строки, создайте одну очередь однозначных чисел, а вторую очередь двузначных чисел. Перемножьте соответ- ственные элементы двух очередей и организуйте третью очередь. Результат выведите на экран. Решение в программе оформляйте через подпрограммы.
10. Используя очередь, проверьте, какие строки являются симметричными. Решение в программе оформляйте через подпрограммы.
11. Используя очередь, проверьте на равенство два текстовых файла. Решение в про- грамме оформляйте через подпрограммы.
12. Создайте две очереди. Проверьте, является ли одна из очередей частью другой. Ре- шение в программе оформляйте через подпрограммы.
13. Создать список с помощью массива целых чисел. Элементы списка в обратном по- рядке вывести на экран.
14. Создать список с помощью массива целых чисел. Все четные элементы списка вы- вести на экран.
15. Создать список с помощью массива целых чисел. Все нечетные элементы списка вывести на экран.
16. Создать односвязный список с помощью массива целых чисел. Отсортировать эле- менты списка по возрастанию, задавая порядок чисел массивом индексов следующих элементов (next). В результате массив чисел остается без изменений, массив индексов пе- реупорядочивается. Результирующий список вывести на экран.
17. Создать односвязный список с помощью массива целых чисел. Расположить в начале списка все четные элементы списка, задавая порядок чисел массивом индексов следующих элементов (next). В результате массив чисел остается без изменений, массив

индексов переупорядочивается. Результирующий список вывести на экран.

1. Создать односвязный список с помощью массива целых чисел. Исключить из спис- ка все нулевые элементы, задавая порядок чисел массивом индексов следующих элемен- тов (next). В результате массив чисел остается без изменений, массив индексов переупо- рядочивается. Результирующий список вывести на экран.
2. Создать односвязный список с помощью массива целых чисел. Исключить из спис- ка все нулевые элементы, задавая порядок чисел массивом индексов следующих элемен- тов (next). В результате массив чисел остается без изменений, массив индексов переупо- рядочивается. Найти сумму все четных элементов списка. Результирующий список и сумму вывести на экран.
3. Создать односвязный список с помощью массива целых чисел. Ввести с клавиату- ры число и поместить его за пятым элементом списка, задавая порядок чисел массивом индексов следующих элементов (next). Результирующий список вывести на экран.
4. Создать односвязный список с помощью массива целых чисел. Ввести с клавиату- ры число и поместить его перед тем элементом списка, который больше него. Результи- рующий список вывести на экран. Порядок чисел в списке задается массивом индексов следующих элементов (next).
5. Создать односвязный список с помощью массива целых чисел. Ввести с клавиату- ры число, найти это число в списке и удалить. Результирующий список вывести на экран. Порядок чисел в списке задается массивом индексов следующих элементов (next).
6. Создать односвязный список с помощью массива целых чисел. Ввести с клавиату- ры число, найти все элементы в этим числом в списке и удалить. Результирующий список вывести на экран. Порядок чисел в списке задается массивом индексов следующих эле- ментов (next).
7. Создать односвязный список с помощью массива целых чисел. Поменять местами четные и нечетные элементы списка (рядом стоящие). Результирующий список вывести на экран. Порядок чисел в списке задается массивом индексов следующих элементов (next).
8. Создать односвязный список с помощью массива целых чисел. Сформировать но- вый список, в котором элементы следуют от конца к началу (последний элемент станет первым, предпоследний – вторым и т.д.). Результирующий список вывести на экран. По- рядок чисел в списке задается массивом индексов следующих элементов (next).
9. Создать односвязный список с помощью массива целых чисел. Продублировать в списке первый, третий и пятый элементы. Результирующий список вывести на экран. Порядок чисел в списке задается массивом индексов следующих элементов (next).
10. Создать односвязный список с помощью массива целых чисел. Удалить в списке первый, третий и пятый элементы.. Результирующий список вывести на экран. Порядок чисел в списке задается массивом индексов следующих элементов (next).
11. Создать список с помощью массива структур. Элементы списка в обратном поряд- ке вывести на экран.
12. Создать список с помощью массива структур. Все четные элементы списка вывести на экран.
13. Создать список с помощью массива структур. Все нечетные элементы списка выве- сти на экран.
14. Создать односвязный список с помощью массива структур. Отсортировать элемен- ты списка по возрастанию. Результирующий список вывести на экран.
15. Создать односвязный список с помощью массива структур. Один из элементов структуры – целое число. Расположить в начале списка все элементы списка с четными целыми числами. Результирующий список вывести на экран.
16. Создать односвязный список с помощью массива структур. Исключить из списка

все элементы с нулевым целым числом. Результирующий список вывести на экран.

1. Создать односвязный список с помощью массива структур. Исключить из списка все элементы с нулевым целым числом. Найти сумму тех целочисленных элементов списка, которые являются четными. Результирующий список и сумму вывести на экран.
2. Создать односвязный список с помощью массива структур. Создать новый элемент списка, ввести с клавиатуры число и поместить его в целочисленное поле нового элемен- та. Разместить новый элемент за пятым элементом списка. Результирующий список вы- вести на экран.
3. Создать односвязный список с помощью массива структур. Создать новый элемент списка, ввести с клавиатуры число и поместить его в целочисленное поле нового элемен- та. Разместить новый элемент перед тем элементом списка, целочисленное поле которого имеет большее значение. Результирующий список вывести на экран.
4. Создать односвязный список с помощью массива структур. Ввести с клавиатуры число, найти это число в списке (в целочисленном поле) и удалить соответствующий элемент списка. Результирующий список вывести на экран.
5. Создать односвязный список с помощью массива структур. Ввести с клавиатуры число, найти все элементы в этим числом (в целочисленном поле) в списке и удалить. Ре- зультирующий список вывести на экран.
6. Создать односвязный список с помощью массива структур. Поменять местами чет- ные и нечетные элементы списка (рядом стоящие). Результирующий список вывести на экран.
7. Создать односвязный список с помощью массива структур. Сформировать новый список, в котором элементы следуют от конца к началу (последний элемент станет пер- вым, предпоследний – вторым и т.д.).). Результирующий список вывести на экран.
8. Создать односвязный список с помощью массива структур. Продублировать в списке первый, третий и пятый элементы. Результирующий список вывести на экран.
9. Создать односвязный список с помощью массива структур. Удалить в списке пер- вый, третий и пятый элементы. Результирующий список вывести на экран.
10. Написать программу, содержащую процедуру, которая меняет местами первый и второй элементы непустого списка. Если элементы не найдены, то выдать на экран соот- ветствующие сообщение.
11. Написать программу, содержащую процедуру, которая меняет местами первый и пятый элементы непустого списка. Если элементы не найдены, то выдать на экран соот- ветствующие сообщение.
12. Написать программу, содержащую процедуру, которая меняет местами первый и последний элементы непустого списка. Если элементы не найдены, то выдать на экран соответствующие сообщение.
13. Написать программу, содержащую процедуру, которая вставляет новый элемент перед каждым вхождением заданного элемента. Если элементы не найдены, то выдать на экран соответствующие сообщение.
14. Написать программу, содержащую процедуру, которая вставляет новый элемент за каждым вхождением заданного элемента. Если элементы не найдены, то выдать на экран соответствующие сообщение.
15. Написать программу, содержащую подпрограмму, которая проверяет на равенство списки М1 и М2.
16. Написать программу, содержащую функцию, которая определяет, входит ли список М1 в список М2. Предполагается, что списки существуют.
17. Написать программу, содержащую подпрограмму, которая копирует в конец непу- стого списка М его первый элемент. Если элементы не найдены, то выдать на экран соот- ветствующие сообщение.
18. Написать программу, содержащую подпрограмму, которая копирует в начало не- пустого списка М его последний элемент. Если элементы не найдены, то выдать на экран соответствующие сообщение.
19. Написать программу, содержащую процедуру, которая копирует в список М за каждым вхождением заданного элемента все элемента списка М1.
20. Написать программу, содержащую процедуру, которая объединяет два упорядо- ченных по не убыванию списка М1 и М2 в один упорядоченный по не убыванию список, построив новый список М.
21. Написать программу, содержащую процедуру, которая объединяет два упорядо- ченных по не убыванию списка М1 и М2 в один упорядоченный по не убыванию список, сменив соответствующим образом ссылки в М1 и М2.
22. Написать программу, содержащую функцию, которая проверяет, упорядочены ли элементы списка по алфавиту.
23. Напишите программу сортировки существующего списка по алфавиту. В програм- ме используйте подпрограммы.
24. Напишите программу, которая создавала бы файл целых чисел, а затем формирова- ла список целых чисел файла. Создайте в конце списка элемент, содержащий сумму всех чисел файла. В программе используйте подпрограммы.
25. Напишите программу, которая создавала бы файл целых чисел, а затем формирова- ла список целых чисел файла. Создайте список чисел, являющихся суммой соседних эле- ментов. В программе используйте подпрограммы.
26. Напишите программу, которая создавала бы текстовый файл, а затем формировала список строк файла. Создайте список обратных строк. В программе используйте подпро- граммы.
27. Напишите программу, которая создавала бы текстовый файл, а затем формировала список строк файла. Создайте отсортированный список строк. В программе используйте подпрограммы.
28. Напишите программу, которая создавала бы файл комбинированного типа, а затем формировала список, используя какое-либо поле записи. Создайте отсортированный спи- сок. В программе используйте подпрограммы.
29. Напишите программу, которая создавала бы файл комбинированного типа, а затем формировала список элементов файла. Создайте отсортированный по какому-либо полю список. В программе используйте подпрограммы.
30. Составить программу, которая вводит с клавиатуры названия городов, динамиче- ски отводит место в памяти под каждое название и строит из них связанный список, упо- рядоченный по алфавиту. По окончании формирования список городов вывести на экран монитора.
31. Составить список учебной группы, содержащий не менее 15 студентов. Указать для каждого студента оценки, полученные на последних четырех экзаменах. Разработать про- грамму, которая вводит данные с клавиатуры о каждом студенте, строит односвязный список, а затем удаляет из списка элементы, относящиеся к неуспевающим студентам.
32. Информацию о величине экспорта и соответствующий номер контракта записать в двухсвязный кольцевой список. Затем переместить данную информацию в двумерный динамический массив и осуществить поиск максимальной величины экспорта. На экран вывести искомую информацию.
33. Создать двухсвязный список, содержащий следующую информацию: год и соот- ветствующую численность населения. Программу организовать таким образом, чтобы на экран выводилась информация, численность населения в которой была больше введен- ного с клавиатуры значения.
34. Ввести с клавиатуры строку символов, формируя из ее элементов двусвязный спи-

сок. Написать программу, которая формирует двусвязный список из входной строки. В поле данных каждого элемента списка записывается отдельный символ. В программе производится анализ первого символа входной строки: если это буква ***'А'***, то в конец списка добавляется еще одна буква ***'А'***, иначе из списка исключаются все буквы ***'А'***. По- лученный результат выводится на экран.

1. Описать не рекурсивную функцию, которая определяет число вхождений данного элемента в двоичное дерево. Тип данных – int
2. Описать не рекурсивную функцию, которая вычисляет сумму элементов двоичного дерева. Тип данных – int.
3. Описать не рекурсивную функцию, которая вычисляет сумму элементов двоичного дерева, которые расположены справа. Тип данных – int.
4. Описать не рекурсивную функцию, которая вычисляет сумму элементов двоичного дерева, которые расположены слева. Тип данных – int.
5. Описать не рекурсивную функцию, которая вычисляет средне значение элементов двоичного дерева. Тип данных – int.
6. Найти минимальное значение и удалить его без потери всего остального узла. Сде- лать не рекурсивно. Тип данных – int.
7. Найти максимальное значение и удалить его без потери всего остального узла. Сделать не рекурсивно. Тип данных – int.
8. Найти минимальное значение и удалить узел с этим элементом. Сделать не рекур- сивно. Тип данных – int.
9. Найти максимальное значение и удалить узел с этим элементом. Сделать не рекур- сивно. Тип данных – int.
10. Найти элемент, введенный с клавиатуры, и если он есть то после него добавить слева этот элемент. Сделать не рекурсивно. Тип данных – int.
11. Найти элемент, введенный с клавиатуры, если он есть удалить все последующие узлы, а этот элемент оставить. Сделать не рекурсивно. Тип данных – int.
12. Найти элемент, введенный с клавиатуры, и удалить его(если он есть) без потери всего остального узла. Сделать не рекурсивно. Тип данных – int.
13. Найти элемент, введенный с клавиатуры, и удалить узел (если он есть) с этим эле- ментом. Сделать не рекурсивно. Тип данных – int.
14. В дерево L1 добавить все элементы дерева L2 не рекурсивно, с изменением ссылок. Соблюдать правило бинарного дерева: слева меньшие значения, справа большие.
15. В дерево L1 добавить все элементы дерева L2 не рекурсивно, с использованием ко- пий элементов дерева L2. L2 не изменяется. Соблюдать правило бинарного дерева: слева меньшие значения, справа большие.
16. Описать не рекурсивную функцию, которая находит длину (количество узлов) на пути от корня дерева до ближайшего узла, содержащего данный элемент (если такого уз- ла в дереве нет, то считать результат равным -1). Тип данных – int.
17. Описать не рекурсивную функцию, которая определяет, является ли данное дере- во деревом двоичного поиска (т.е. по отношению к любому узлу в этом дереве его левое поддерево содержит только те данные, значения которых меньше значения данного узла, а его правое поддерево содержит только те данные, значения которых больше значения данного узла). Тип данных – int.
18. Описать не рекурсивную функцию, которая подсчитывает количество узлов на N- ом уровне непустого двоичного дерева (корень считать узлом нулевого уровня). Тип дан- ных – int.
19. Описать не рекурсивную функцию, которая находит максимальное значение на N- ом уровне непустого двоичного дерева (корень считать узлом нулевого уровня). Тип дан- ных – int.
20. Описать не рекурсивную функцию, которая находит наименьшее значение на N-ом уровне непустого двоичного дерева (корень считать узлом нулевого уровня). Тип данных

* int.

1. Описать рекурсивную функцию, которая подсчитывает количество узлов на N-ом уровне непустого двоичного дерева (корень считать узлом нулевого уровня). Тип данных

* int.

1. Описать не рекурсивную функцию, которая печатает все элементы двоичного де- рева по уровням, начиная с корня, на каждом уровне – слева направо. Тип данных – int.
2. Описать рекурсивную функцию, которая вычисляет сумму элементов двоичного дерева, которые расположены справа. Тип данных – int.
3. Описать рекурсивную функцию, которая вычисляет сумму элементов двоичного дерева, которые расположены слева. Тип данных – int.
4. Описать рекурсивную функцию, которая вычисляет средне значение элементов двоичного дерева. Тип данных – int.
5. Найти минимальное значение и удалить его без потери всего остального узла. Сде- лать рекурсивно. Тип данных – int.
6. Найти максимальное значение и удалить его без потери всего остального узла. Сделать н рекурсивно. Тип данных – int.
7. Найти минимальное значение и удалить узел с этим элементом. Сделать н рекур- сивно. Тип данных – int.
8. Найти максимальное значение и удалить узел с этим элементом. Сделать н рекур- сивно. Тип данных – int.
9. Описать рекурсивную функцию, которая печатает все элементы двоичного дерева по уровням, начиная с корня, на каждом уровне – слева направо. Тип данных – int.
10. Описать рекурсивную функцию, которая определяет число вхождений данного элемента в двоичное дерево. Тип данных – char\*.
11. Описать не рекурсивную функцию, которая проверяет, идентичны ли два двоичных дерева. Тип данных – char\*
12. Найти элемент, введенный с клавиатуры, и удалить его(если он есть) без потери всего остального узла. Сделать рекурсивно. Тип данных – int.
13. Найти элемент, введенный с клавиатуры, и удалить узел (если он есть) с этим эле- ментом. Сделать рекурсивно. Тип данных – int.
14. Найти элемент, введенный с клавиатуры, и если он есть то после него добавить слева этот элемент. Сделать рекурсивно. Тип данных – int.
15. Найти элемент, введенный с клавиатуры, если он есть удалить все последующие узлы, а этот элемент оставить. Сделать рекурсивно. Тип данных – int.
16. Описать рекурсивную функцию, которая вычисляет сумму элементов двоичного дерева. Тип данных – char\*.
17. Описать рекурсивную функцию, которая определяет, входит ли данный элемент в двоичное дерево. Тип данных – char\*.
18. Описать рекурсивную функцию, которая печатает значения данных из всех узлов дерева, не являющихся листьями. Тип данных – char\*.
19. Описать рекурсивную функцию, которая проверяет, идентичны ли два двоичных дерева. Тип данных – char\*
20. Описать функцию, которая в дерево двоичного поиска вставляет новый элемент Тип данных – char\*.
21. В дерево L1 добавить все элементы дерева L2 рекурсивно, с изменением ссылок. Соблюдать правило бинарного дерева: слева меньшие значения, справа большие.
22. В дерево L1 добавить все элементы дерева L2 рекурсивно, с использованием копий элементов дерева L2. L2 не изменяется. Соблюдать правило бинарного дерева: слева меньшие значения, справа большие.
23. В текстовом файле найти и вывести k-ое слово (k - вводится с клавиатуры, доп. массивов не использовать).
24. В текстовом файле найти самое короткое слово и вывести на экран (доп. массивов не использовать.
25. Отсортировать двунаправленный список по возрастанию. Производить сортировку указателей, а не значений.
26. В бинарном файле удалить все числа большие k. k- вводится с клавиатуры. Допол- нительных массивов не использовать.
27. В бинарном файле удалить все числа, стоящие после максимального. Дополнитель- ных массивов не использовать.
28. В текстовом файле найти сколько раз встречается введенная последовательность символов. Дополнительных массивов не использовать.
29. В текстовом файле поменять местами k-ое и m-ое слова. k, m- вводятся с клавиату- ры. Дополнительных массивов не использовать.
30. В текстовом файле удалить введенное слово. Дополнительных массивов не исполь- зовать.
31. В текстовом файле удалить лишние пробелы. Дополнительных массивов не ис- пользовать. (Между словами и после знаков препинания должен быть один пробел)
32. Даны три файла целых чисел одинакового размера с именами NameA, NameB и NameC. Создать новый файл с именем NameD, в котором чередовались бы элементы ис- ходных файлов с одним и тем же номером: A0, B0, C0, A1, B1, C1, A2, B2, C2, ...
33. Для существующего упорядоченного бинарного файла ввести с клавиатуры не- сколько чисел, размещая их в файле без нарушения его упорядоченности по возрастанию. Дополнительных массивов и файлов не использовать.
34. Есть два текстовых файла отсортированных по убыванию. Без выполнения сорти- ровки записать числа из этих файлов в третий файл по возрастанию.
35. Есть два текстовых файла. В первом числа отсортированы по возрастанию, во вто- ром по убыванию. Без выполнения сортировки записать числа из этих файлов в третий файл по возрастанию.
36. Из введенной последовательности чисел создать полное сбалансированное дерево. (если чисел для создания не хватает, запросить у пользователя дополнительные числа)
37. Имеются два бинарных файла, упорядоченных по возрастанию. Переписать ин- формацию в третий файл, упорядочивая ее по убыванию. Дополнительных массивов и файлов не использовать, сортировку не выполнять
38. Компоненты файла f – целые двухзначные числа (положительные и отрицатель- ные). Получить файл g, образованный из f включением только чисел кратных K.
39. Найти сколько раз в текстовом файле встречается k-ое слово. (k - вводится с клави- атуры, доп. массивов не использовать).
40. Написать не рекурсивную функцию создания бинарного дерева.
41. Написать рекурсивную функцию создания бинарного дерева.
42. Написать функцию для удаления введенного значения из двунаправленного списка. Дополнительных списков и массивов не использовать.
43. Написать функцию для удаления введенного значения из однонаправленного спис- ка. Дополнительных списков и массивов не использовать.
44. Написать функцию для удаления всех нечетных значений из однонаправленного списка.
45. Написать функцию для удаления всех четных значений из двунаправленного спис- ка.
46. Написать функцию для удаления из двунаправленного списка элементов больше введенного. Дополнительных списков и массивов не использовать.
47. Написать функцию для удаления из двунаправленного списка элементов меньших введенного. Дополнительных списков и массивов не использовать.
48. Написать функцию для удаления из однонаправленного списка элементов больших введенного. Дополнительных списков и массивов не использовать.
49. Написать функцию для удаления из однонаправленного списка элементов меньших введенного. Дополнительных списков и массивов не использовать.
50. Написать функцию для удаления каждого k-ого числа в двунаправленном списке.
51. Отсортировать однонаправленный список по возрастанию.
52. Создать текстовый файл F1 не менее, чем из 10 строк и записать в него информа- цию. Скопировать в файл F2 только те строки из F1, которые начинаются с буквы «А».
53. Создать текстовый файл F1 не менее, чем из 10 строк и записать в него информа- цию. Скопировать из файла F1 в файл F2 строки, начиная с К до К+5.
54. Удалить в бинарном файле каждое k-ое число. k- вводится с клавиатуры. Дополни- тельных массивов не использовать.
55. Удалить в бинарном файле каждую k-ю последовательность из m чисел. k, m- вво- дятся с клавиатуры. Дополнительных массивов не использовать.
56. Удалить из бинарного дерева введенное число.
57. Удалить из бинарного дерева все элементы ниже 3 уровня.
58. Удалить из бинарного файла числа, содержащиеся в текстовом файле. В файлах со- держатся только целые числа.
59. Удалить из дерева все листья
60. Удалить из дерева все четные числа
61. Написать рекурсивную функцию поиска узла бинарного дерева с максимальным значением (число с плавающей точкой). Ключ дерева – число с фиксированной точкой.
62. Написать рекурсивную функцию поиска узла бинарного дерева с найменьшим значением (число с плавающей точкой). Ключ дерева – число с фиксированной точкой.
63. Дано бинарное дерево. Записи бинарного дерева содержат указатель на функцию и указатель на стек, хранящий строку знаков. Одна из функций определяет правильность расстановки скобок в строке. Вторая функция переставляет самую длинную и самую ко- роткую строки в соответствующих стеках.
64. Дан массив бинарных деревьев. Элементеми дерева являются фамилия, оценки, ад- рес, указатель на текст с характеристикой студента, признак – военнообязанный или нет. Сделать копии бинарных деревьев с целью воссоздания информации, если исходные де- ревья и вся принадлежащая им информация будут уничтожены. Структура новых деревь- ев должна полностью совпадать со структурами исходных деревьев. Рекурсивные функ- ции и глобальные переменные не использовать.
65. Дана ссылка на бинарное дерево, записи которого содержат указатель на стек, в ко- тором посимвольно хранится строка, являющаяся ключом бинарного дерева. Сформиро- вать кольцо из двадцати пяти записей,в которых находятся первые в алфавитном поряд- кеключи бинарного дерева (ключ не более десяти знаков). Указатели на стеки, содержа- щие выбранные строки (ключи бинарного дерева), записать в файл. Имя файла ввести. Рекурсии и системные функции не использовать.
66. Дан стек. Элементами стека являются число N и указатель на бинарное дерево. В узле бинарного дерева записаны указатели на строку и на массив целых чисел. На N-м уровне каждого дерева найти его средний элемент (узел) и , используя рекурсию, на ста- ром месте записать строку знаков в обратном порядке. Рекурсивно вывести строки с хво-

ста. Узлов на одном уровне не более пятидесяти. Глобальные переменные не использо- вать.

1. Дано бинарное дерево, в качестве ключа используется строка знаков (длина не бо- лее десяти знаков). В записях дерева хранятся указатели на кольцо. В записях кольца есть указатели на строки знаков (не более ста знаков). Слова в строке разделены произволь- ным количеством пробелов. Определить частоту встречаемости слов во всех строках кольца. Вывести в файл слова и соответствующую им частоту. Имя файла для каждого кольца задается с клавиатуры. Глобальные переменные, системные функции не исполь- зовать, исходные строки сохранить. Самое длинное слово не более восьми знаков.
2. Дано бинарное дерево, в качестве ключа используется строка, ее длина не более пяти знаков. Записи дерева содержат указатель на слово, его длина не более пятнадцати знаков. В поддереве минимальной длины определить самое длинное слово.
3. Используя стек, напечатать символы некоторой величины строкового типа в обрат- ном порядке.
4. Даны три стека. Два первых стека пустые, в другом находятся натуральные числа. В первый пустой стек поместить четные числа, взятые из непустого стека, во второй пу- стой стек поместить нечетные числа так же взятые их непустого стека.
5. В математическом выражении встречаются скобки трёх типов: круглые (), квадрат- ные [] и фигурные {}. Необходимо проверить, выполняется ли баланс скобок.
6. Отсортировать однонаправленный список по возрастанию. Производить сортиров- ку указателей, а не значений.