

Министерство образования и науки Российской Федерации
Федеральное государственное образовательное учреждение высшего
профессионального образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра комплексной информационной безопасности
электронно-вычислительных систем (КИБЭВС)

Е.М. Давыдова

Методические указания к практическим занятиям по дисциплине
«Методы программирования».
Для специальности 090105 Комплексное обеспечение информационной безопасности ав-
томатизированных систем

2012

Содержание

1. Алгоритмизация.....	3
1.1 Цель работы	3
1.2 Теоретические сведения	3
1.3 Задание к самостоятельной работе.	5
2. Структурное тестирование программного обеспечения	5
2.1.Цель работы	5
2.2 Теоретические сведения	5
2.3 Задание к самостоятельной работе	8
3. Характеристики качества программных средств	8
3.1 Цель работы	8
3.2 Краткие теоретические сведения	8
3.3 Задание к самостоятельной работе	10
4. Анализ рисков.....	10
4.1 Цель работы	10
4.3. Задание к самостоятельной работе	12
ПРИЛОЖЕНИЕ 1 Варианты заданий.....	13
ПРИЛОЖЕНИЕ 2 Требования к оформлению отчетов	28

1. Алгоритмизация

1.1 Цель работы

Научиться описывать программы.

1.2 Теоретические сведения

Описание программы должно содержать следующие разделы:

- общие сведения;
- функциональное назначение;
- описание логической структуры;
- используемые технические средства;
- вызов и загрузка;
- входные данные;
- выходные данные.

В зависимости от особенностей программы допускается вводить дополнительные разделы или объединять отдельные разделы.

В разделе «Общие сведения» должны быть указаны:

- обозначение и наименование программы;
- программное обеспечение, необходимое для функционирования программы;
- языки программирования, на которых написана программа.

В разделе «Функциональное назначение» должны быть указаны классы решаемых задач и (или) назначение программы и сведения о функциональных ограничениях на применение.

В разделе «Описание логической структуры» должны быть указаны:

- алгоритм программы;
- используемые методы;
- структура программы с описанием функций составных частей и связи между ними;
- связи программы с другими программами.

Описание логической структуры программы выполняется с учетом текста программы на исходном языке.

В разделе «Используемые технические средства» должны быть указаны типы электронно-вычислительных машин и устройств, которые используются при работе программы.

В разделе «Вызов и загрузка» должны быть указаны:

- способ вызова программы с соответствующего носителя данных;
- входные точки в программу.

Допускается указывать адреса загрузки, сведения об использовании оперативной памяти, объем программы.

В разделе «Входные данные» должны быть указаны:

- характер, организация и предварительная подготовка входных данных;
- формат, описание и способ кодирования входных данных.

В разделе «Выходные данные» должны быть указаны:

- характер и организация выходных данных;
- формат, описание и способ кодирования выходных данных.

Пусть необходимо решить следующую задачу:

Задан одномерный массив A размерностью N . Найти максимальный элемент массива, результат решения отобразить на экране.

При составлении алгоритма необходимо обозначить, и объяснить назначение и формат всех используемых переменных.

MAX– имя переменной, I - счетчик циклов.

Алгоритм можно описать в любом виде.

Описание алгоритма А)

Переменной MAX присвоить значение первого элемента массива. Затем значение MAX поочередно сравнить с оставшимися элементами массива, и если значение переменной MAX будет меньше очередного элемента массива ($A[I]$), то переменной MAX присвоить это значение. Просмотр последовательности элементов продолжается до тех пор, пока не будут перебраны все элементы массива A. Найденное значение переменной MAX и будет искомым.

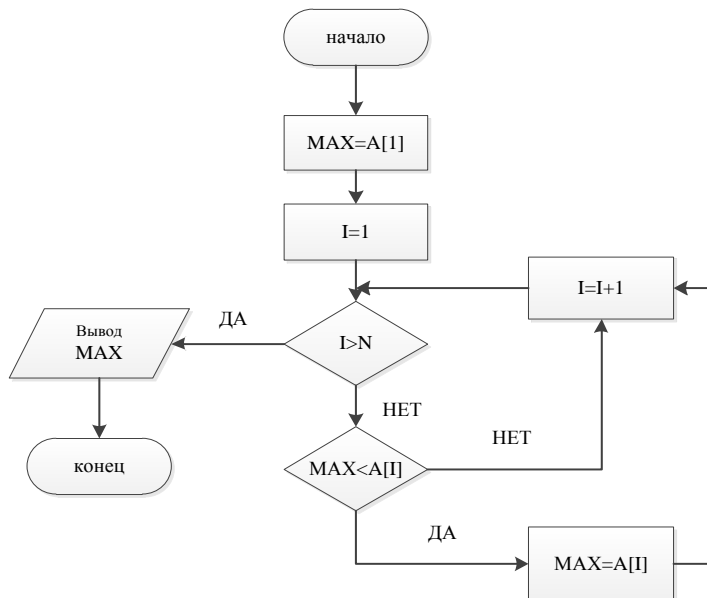


Рисунок 1 – Пример схемы алгоритма

Описание алгоритма Б)

Шаг 0.Начало.

Шаг 1.Переменной MAX присвоить значение первого элемента массива $A[1]$.

Шаг 2.Счетчику циклов I присвоить значение 1.

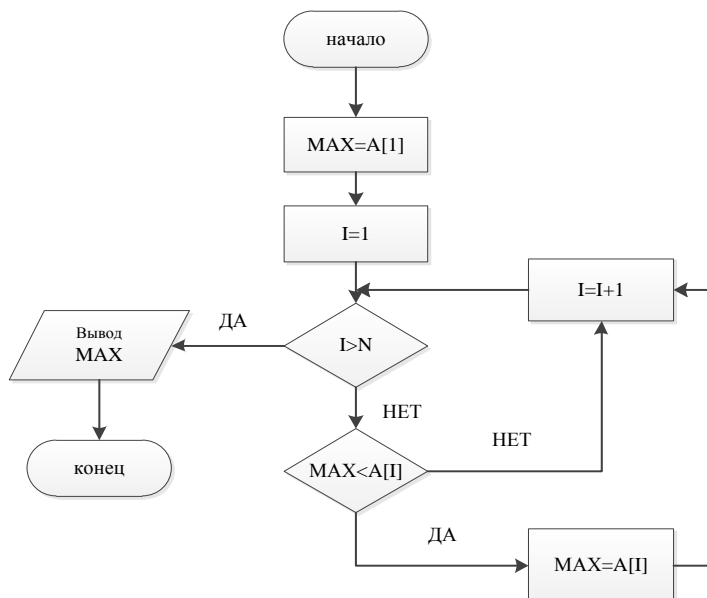
Шаг 3.Если $I > N$, то переход к шагу 5.

Шаг 4.Сравнить значение переменной MAX с очередным значением из массива $A[I]$. Если значение MAX больше, чем значение $A[I]$, то переход к шагу 5. В противном случае – переменной MAX присваивают значение $A[I]$. Переход к шагу 5.

Шаг5. Увеличить значение счетчика I на 1. Переход к шагу 3.

Шаг 5.Вывод на печать значения MAX.

Шаг 6.Конец.



1.3 Задание к самостоятельной работе.

- Изучить ГОСТ 19 XXX.
- получить задание у преподавателя на разработку программ (ПРИЛОЖЕНИЕ 1).
- В соответствии с ГОСТ 19 XXX описать проекты программ в соответствии с выданным вариантом самостоятельной работы и разработать для них схемы алгоритмов.
- Написать отчет. Требования к отчету приведены в ПРИЛОЖЕНИИ 2
- Защитить выполненную работу.

2. Структурное тестирование программного обеспечения

2.1. Цель работы

Научиться тестировать программные системы

2.2 Теоретические сведения

Основные понятия и принципы тестирования ПО.

Тестирование — процесс выполнения программы с целью обнаружения ошибок.

Шаги процесса задаются тестами. Каждый тест определяет:

- свой набор исходных данных и условий для запуска программы;
- набор ожидаемых результатов работы программы.

Другое название теста — тестовый вариант. Полную проверку программы гарантирует исчерпывающее тестирование. Оно требует проверить все наборы исходных данных, все варианты их обработки и включает большое количество тестовых вариантов. Исчерпывающее тестирование во многих случаях затруднительно, поскольку срабатывают ресурсные ограничения (прежде всего, ограничения по времени).

Хорошим считают тестовый вариант с высокой вероятностью обнаружения еще не раскрытой ошибки. Успешным называют тест, который обнаруживает до сих пор не раскрытую ошибку.

Целью проектирования тестовых вариантов является систематическое обнаружение различных классов ошибок при минимальных затратах времени и стоимости.

На входе процесса тестирования три потока:

- текст программы;

- исходные данные для запуска программы;
- ожидаемые результаты.

Выполняются тесты, все полученные результаты оцениваются. Это значит, что реальные результаты тестов сравниваются с ожидаемыми результатами. Когда обнаруживается несовпадение, фиксируется ошибка — начинается отладка. Процесс отладки непредсказуем по времени. На поиск места дефекта и исправление может потребоваться час, день, месяц. Неопределенность в отладке приводит к большим трудностям в планировании действий.

После сбора и оценивания результатов тестирования начинается отображение качества и надежности ПО. Если регулярно встречаются серьезные ошибки, требующие проектных изменений, то качество и надежность ПО подозрительны, констатируется необходимость усиления тестирования.

Результаты, накопленные в ходе тестирования, могут оцениваться и более формальным способом. Для этого используют модели надежности ПО, выполняющие прогноз надежности по реальным данным об интенсивности ошибок.

Существуют два принципа тестирования программы:

1. Функциональное тестирование (тестирование «черного ящика»). Известны: функции программы. Исследуется - работа каждой функции на всей области определения.
2. Структурное тестирование (тестирование «белого ящика»). Известна: внутренняя структура программы. Исследуются: внутренние элементы программы и связи между ними.

Структурное тестирование.

Тестирование базового пути — это способ, который основан на принципе «белого ящика».

Способ тестирования базового пути дает возможность:

- получить оценку комплексной сложности программы;
- использовать эту оценку для определения необходимого количества тестовых вариантов.

Тестовые варианты разрабатываются для проверки базового множества путей (маршрутов) в программе. Они гарантируют однократное выполнение каждого оператора программы при тестировании.

Для представления программы используется потоковый граф. Перечислим его особенности.

1. Граф строится отображением управляющей структуры программы. В ходе отображения закрывающие скобки условных операторов и операторов циклов (end if; end loop) рассматриваются как отдельные (фиктивные) операторы.

2. Узлы (вершины) потокового графа соответствуют линейным участкам программы, включают один или несколько операторов программы.

3. Дуги потокового графа отображают поток управления в программе (передачи управления между операторами). Дуга — это ориентированное ребро.

4. Различают операторные и предикатные узлы. Из операторного узла выходит одна дуга, а из предикатного — две дуги.

5. Предикатные узлы соответствуют простым условиям в программе. Составное условие программы отображается в несколько предикатных узлов. Составным называют условие, в котором используется одна или несколько булевых операций (OR, AND).

Замкнутые области, образованные дугами и узлами, называют регионами.

Окружающая граф среда рассматривается как дополнительный регион.

Цикломатическая сложность — метрика ПО, которая обеспечивает количественную оценку логической сложности программы. В способе тестирования базового пути Цикломатическая сложность определяет:

- количество независимых путей в базовом множестве программы;
- верхнюю оценку количества тестов, которое гарантирует однократное выполнение всех операторов.

Независимым называется любой путь, который вводит новый оператор обработки или новое условие. В терминах потокового графа независимый путь должен содержать дугу, не входящую в ранее определенные пути.

Путь начинается в начальном узле, а заканчивается в конечном узле графа. Независимые пути формируются в порядке от самого короткого к самому длинному.

Свойства базового множества:

1. Тесты, обеспечивающие его проверку, гарантируют:
 - однократное выполнение каждого оператора;
 - выполнение каждого условия по True-ветви и по False-ветви;
2. мощность базового множества равна цикломатической сложности потокового графа.

Значение 2-го дает априорную оценку количества независимых путей, которое имеет смысл искать в графе.

Цикломатическая сложность вычисляется одним из трех способов:

- 1) цикломатическая сложность равна количеству регионов потокового графа;
- 2) цикломатическая сложность определяется по формуле

$V(G) = E - N + 2$, где E — количество дуг, N — количество узлов потокового графа;

- 3) цикломатическая сложность формируется по выражению $V(G) = p + 1$, где p — количество предикатных узлов в потоковом графе G .

Шаги способа тестирования базового пути

Шаг 1. На основе текста программы формируется потоковый граф.

- нумеруются операторы текста (номера операторов показаны в тексте процедуры);
- производится отображение пронумерованного текста программы в узлы и вершины потокового графа.

Шаг 2. Определяется цикломатическая сложность потокового графа — по каждой из трех формул:

- 1) $V(G) = 6$ регионов;
- 2) $V(G) = 17$ дуг - 13 узлов + 2 = 6;
- 3) $V(G) = 5$ предикатных узлов + 1 = 6.

Шаг 3. Определяется базовое множество независимых линейных путей.

Для удобства дальнейшего анализа по каждому пути указаны условия запуска..

Шаг 4. Подготавливаются тестовые варианты, инициирующие выполнение каждого пути.

Каждый тестовый вариант формируется в следующем виде: Исходные данные (ИД): Ожидаемые результаты (ОЖ.РЕЗ.):

Исходные данные должны выбираться так, чтобы предикатные вершины обеспечивали нужные переключения — запуск только тех операторов, которые перечислены в конкретном пути, причем в требуемом порядке.

Реальные результаты каждого тестового варианта сравниваются с ожидаемыми результатами. После выполнения всех тестовых вариантов гарантируется, что все операторы программы выполнены по меньшей мере один раз.

Важно отметить, что некоторые независимые пути не могут проверяться изолированно. Такие пути должны проверяться при тестировании другого пути (как часть другого тестового варианта).

2.3 Задание к самостоятельной работе

1. Изучить методы тестирования белого и черного ящиков.
2. Получить у преподавателя задание на тестирование. Выбрать программу
3. Построить потоковый граф.
4. Построить тесты к программе.
5. Проверить программу.
6. Выявить неточности и некорректности в программе.
7. Исправить программу.
8. Написать отчет и защитить у преподавателя.

3. Характеристики качества программных средств

3.1 Цель работы

Целью работы является изучение и определение характеристик качества программного средства.

3.2 Краткие теоретические сведения

При разработке программных средств (ПС) формируются требования к его характеристикам, обеспечивающим качество функционирования выпускаемого продукта. Под качеством функционирования понимается множество свойств, обуславливающих пригодность ПС обеспечивать надежное и своевременное представление требуемой информации потребителю для ее дальнейшего использования по назначению. То есть качество ПС можно определить как его пригодность и удобство для решения тех задач, для которых оно разрабатывалось.

Поскольку понятия «удобство», «пригодность» определить достаточно сложно, то и целостного и четкого определения качества ПС не существует. Поэтому в соответствии с особенностями, назначением и свойствами каждого ПС выбирается перечень и значения характеристик качества соответствующие выбранной модели качества.

Основой формального регламентирования показателей качества служит международный стандарт: ISO 9126. В нем описаны: модель качества, внешние метрики качества, внутренние метрики качества, метрики качества в использовании.

Модель состоит из шести групп базовых показателей, каждая из которых может быть детализирована несколькими нормативными субхарактеристиками.

Функциональная пригодность

В контракте и техническом задании для каждого проекта, она должна быть выделена и формализована для однозначного понимания и оценивания всеми партнерами на каждом этапе жизненного цикла (ЖЦ) и при значительных модификациях задач ПС. В силу своей специфичности, при последующем изложении функциональная пригодность обозначается как основная цель и главная характеристика для всего множества типов ПС.

детализируется:

- пригодностью для применения. Эта характеристика обозначает способность программного средства решать определенный круг задач;
- корректностью (правильностью, точностью). Эта характеристика определяет точность вычислений программной системы;
- защищённостью – безопасность функционирования комплекса программ и возможный риск от его применения для людей, бизнеса, и внешней среды;
- способностью к взаимодействию с другими программными системами.

Надежность программного средства характеризуется:

- уровнем завершенности программного средства;
- восстанавливаемостью;

- доступностью, готовностью выполнять предназначенные функции.

Эффективность отражается

- временной эффективностью,
- используемостью ресурсов.

Применимость (практичность) предлагается описывать:

- понятностью;
- простотой использования;
- изучаемостью;
- привлекательностью.

Сопровождаемость представляется:

- удобством для анализа;
- изменяемостью;
- стабильностью;
- тестируемостью.

Переносимость предлагается отражать:

- адаптируемостью;
- простотой установки (инсталляции);
- замещаемостью;
- сосуществованием- соответствием.

Каждая из этих характеристик сопровождается субхарактеристикой – **согласованность**, которая отражает отсутствие противоречий со стандартами и нормативными документами.

Модель представляет собой (рисунок 2)

Все внутренние и внешние характеристики могут быть структурированы в три группы, к которым применимы разные категории метрик (см. шкалирование)

- категориальные – описательные(номинальные метрики) адекватны Функциональные возможности программной системы (ПС)

- количественные метрики применимы для измерения характеристик Надежности, Эффективности сложных комплексов программ;

- качественные метрики в наибольшей степени соответствует Практичности, Сопровождаемости, Мобильности.

При рассмотрении метрик качества использования основными характеристиками качества ПС являются:

- системная эффективность применения программного продукта по назначению;

- продуктивность – производительность при решении основных задач ПС, достигаемая при реально ограниченных ресурсах в конкретной вычислительной и внешней среде применения;

- удовлетворенность требований и затрат пользователей в соответствии с целями при применении ПС по назначению;

- защищенность - безопасность функционирования комплекса программ и возможный риск от его применения для людей, бизнеса, и внешней среды.

Таблицы метрик качества содержат следующие рубрики:

- имя и назначение метрики;

- метод ее применения

- способ измерения или оценивания, формула или вычисляемые элементы данных;

- интерпретация измеряемой величины;

- тип шкалы метрики;

- тип измеряемой величины – размер, время, свойство или структура;

- исходные данные для измерения и сравнения;

- этапы ЖЦ ПС, к которым применима метрика;

- оценка полезности метрики в конкретном проекте ПС

Эту таблицу рекомендуется применять при формировании требований к конкретным внешним и внутренним характеристикам качества в составе спецификаций на функциональные компоненты ПС.

3.3 Задание к самостоятельной работе

1. Изучить характеристики качества программных средств и способы их измерения.
2. Получить у преподавателя задание (программную систему, для которой необходимо определить характеристики качества).
3. Выбрать критерии (определить шкалы измерения).
4. Определить характеристики у заданной программной системы.
5. Написать отчет и защитить у преподавателя.

4. Анализ рисков

4.1 Цель работы

1. Изучить понятие - риск.
2. Научиться выявлять риски.
3. Научиться управлять рисками

4.2. Основные теоретические сведения

Процесс управления рисками состоит из следующих шагов и действий:

- выявление рисков;
- выявление рисков требующих вмешательства;
- разработка снижения риска.

Для выявления рисков, необходимо заполнить следующую таблицу:

Идентификатор	Ис точник	Усло- вия возникно- вения	И ероят- ность	П ослед- ствия	П лия- ние	К онтекст	Свя- занные риски

Перечень возможных источников рисков взять из лабораторной работы по рискам 5 семестра дисциплины «Методы программирования».

Идентификатор – уникальное название, идентифицирующее риск.

Источник – может быть идентифицирован по предметной области (разработка ПО, внедрение и т.д.) или по фактору (соответствие требованиям заказчика, стабильность и работоспособность группы и т.п.)

Условия возникновения – описание условий, при которых риск может реализоваться и поставить проект под удар.

Вероятность – оценка вероятности риска.

Последствия – оценка последствий реализации риска для проекта. Может быть выражена числом в 5- ти или 10 – ти балльной шкале. Показывает относительную важность проблемы. (все последствия должны быть обоснованы)

Влияние – сводная количественная оценка риска для проекта, равная произведению его вероятности на количественную оценку его последствий.

Контекст – дополнительная информация.

Связанные риски – перечень рисков связанных с данным.

Проранжировав риски по степени влияния, необходимо выработать план действий на случай реализации рисков.

План действий предусматривает рассмотрение каждого риска и ответа на вопросы:

1. Достаточно ли информации о данном риске?
2. Может ли группа игнорировать последствия риска и не принимать никаких действий?

3. Может ли группа сделать что-нибудь, чтобы снизить воздействия риска?

4. Можно ли избежать риска?

Выявив риски, нуждающиеся в реагировании, необходимо по возможности:

- снизить вероятность возникновения риска;
- уменьшить размеры потерь;
- изменить последствия риска.

Необходимо разработать стратегию на случай чрезвычайных обстоятельств. В результате, на каждом этапе выполнения проекта необходимо иметь сводный документ

В котором отражено:

- идентификатор риска;
- формулировка риска;
- стратегия управления риском;
- метрики стратегии управления рисками:
 - вероятность;
 - последствия;
 - влияние;
- действия;
- сроки;
- ответственные лица;
- чрезвычайная стратегия;
- пороговые значения и параметры чрезвычайных ситуаций.

Методы обработки мнений экспертов

Предположим, что эксперты оценивают альтернативы в числовых шкалах.

Пусть $g_j(x_i)$ оценка i альтернативы j экспертом ($i=1, \dots, m, j=1 \dots n$). Оценки $g_1(x_i), \dots, g_n(x_i)$ можно рассматривать как «измерения» искомой «истинной характеристики» $g(x_i)$, считая отклонения $g_j(x_i) - g(x_i)$ случайными величинами.

В качестве приближения можно использовать некоторую статистику $\hat{g}(x_i) = \hat{g}(g_1(x_i), \dots, g_n(x_i))$. Обычно это выборочное среднее $\hat{g}(x_i) = \frac{1}{n} \sum_{j=1}^n g_j(x_i)$

Можно использовать другие статистики.

Сложнее, когда альтернативу нельзя оценить одним числом и экспертам предлагаются дать оценки по каждому показателю. Например оценка качества программного изделия складывается из оценок признаков функциональных, экономических, эстетических, эргономических и др.

В этом случае имеем набор чисел $g_{jk}(x_i)$, где k номер- номер признака. Кроме этих чисел экспертов просят оценить степень важности λ_{jk} каждого показателя (если это не выполнено другим способом). Тогда

$$\hat{g}(x_i) = \frac{1}{n} \sum_{j=1}^n \sum_k \lambda_{jk} g_{jk}(x_i)$$

Следующее уточнение вводят в случае неоднородности группы экспертов. Естественно придать различные (а не одинаковые) веса мнениям экспертов, имеющих разную квалификацию. Определение коэффициента α_j компетентности j -го эксперта можно поручить самим экспертам. Пусть каждый из них оценивает компетентность других числами $0 \leq \alpha_{ij} \leq 1$ Усреднение дает $\alpha_j = \sum_i (\alpha_{ij} / \sum_s \alpha_{is})$

В результате получают итоговую оценку

$$\hat{g}(x_i) = \frac{1}{n} \sum_{j=1}^n \sum_k \alpha_j \lambda_{jk} g_{jk}(x_i).$$

В тех случаях, когда эксперты лишь упорядочивают альтернативы, т.е. используют только порядковую шкалу, возможность арифметических операций отпадает. Тогда переходят к обработке либо относительных частот предпочтений данной альтернативы, либо рангов. Иногда используют «медианную» альтернативу или расстояния между ранжированиями.

4.3. Задание к самостоятельной работе

1. Изучить теоретический материал
2. Проанализировать риски для своей курсовой работы (в том числе и риски связанные с потерей или искажением информации при внедрении вышей системы заказчику).
3. Написать отчет и защитить у преподавателя.

ПРИЛОЖЕНИЕ 1 Варианты заданий

Вариант 1

Задание: Разработать алгоритмы для решения задач.

2. На плоскости заданы n квадратов. Каждый из них задан координатами центра X и Y и длиной стороны D . Стороны квадратов ориентированы параллельно осям координат. Найти номера всех тех квадратов, которые не имеют ни одного пересечения (наложения) ни с каким другим квадратом.

3. Состав студентов факультета с разбивкой на группы. Количество специальностей на факультете, групп каждой специальности и студентов в каждой группе задать самостоятельно. Составить модуль поиска адреса (ссылки) элемента списка по его информационным полям.

5. Используя очередь, решить следующую задачу. Содержимое текстового файла f , разделенное на строки, переписать в текстовый файл g , перенося при этом в конец каждой строки все входящие в нее цифры (с сохранением исходного порядка как среди цифр, так и среди остальных литер строки).

6. Написать процедуру или функцию, определяющую максимальную глубину непустого дерева E , т.е. число ветвей в самом длинном из путей от корня дерева до листьев.

7. Решить задачу раскраски графа.

Практические работы по курсу

«Методы программирования»

Вариант 2

Задание: Разработать алгоритмы для решения задач.

2. Задана последовательность натуральных чисел из диапазона $[1, 2147483647]$. Количество чисел в этой последовательности не превышает 100000. Необходимо определить, можно ли выстроить эти числа в отрезок арифметической прогрессии. При необходимости для построения прогрессии порядок чисел в последовательности можно изменять.

Например, из чисел последовательности

9 6 15 21 12 18

МОЖНО построить арифметическую прогрессию, а из чисел последовательности

12456789 НЕЛЬЗЯ построить прогрессию.

3. Список внутренних телефонов организации с разбивкой по отделам. Количество отделов и телефонов внутри отдела задать самостоятельно. Составить модуль поиска всех телефонов с заданными двумя первыми цифрами.

5. Используя очередь, решить следующую задачу.

TYPE имя = (Анна,..., Яков);

дети = ARRAY[имя, имя] OF BOOLEAN; потомки = FILE OF имя;

Считая заданным имя I и массив D типа дети ($D[X, Y] = \text{TRUE}$, если человек по имени Y является ребенком человека по имени X), записать в файл P типа потомки имена всех потомков человека с именем I в следующем порядке: сначала - имена всех его детей, затем - всех его внуков, затем - правнуков и т.д.

6. В непустом дереве T найти длину пути (число ветвей) от корня до ближайшей вершины с элементом E . Если элемент E не входит в T , то за ответ принять "-1".

7. Найти кратчайший путь из одной вершины в другую (поиск в ширину).

Практические работы по курсу

«Методы программирования»

Вариант 3

Задание: Разработать алгоритмы для решения задач.

2. Задан числовой массив A из n элементов. Ступенькой назовем такие элементы $A[i]$, $A[i+1]$, ..., $A[j]$, что:

1) $A[k] < A[k+1]$, $k=i, i+1, \dots, j-1$;
2) $A[i-1] \geq A[i]$, если $i > 1$ и $A[j] \geq A[j+1]$, если $j < n$. Высотой ступеньки назовем разность $A[j]-A[i]$, а длиной - количество элементов массива, входящих в ступеньку. Вычислить:

- 1) высоту и длину каждой из ступенек;
- 2) высоту самой длинной ступеньки;
- 3) длину самой высокой ступеньки.

3. Список внутренних телефонов организации с разбивкой по отделам. Количество отделов и телефонов внутри отдела задать самостоятельно. Составить модуль поиска всех телефонов с заданными двумя первыми цифрами.

5. Используя стек, решить следующую задачу.

В текстовом файле LOG записано без ошибок логическое выражение (ЛВ) в следующей форме:

$\langle \text{ЛВ} \rangle ::= \text{true} \mid \text{false} \mid (\neg \langle \text{ЛВ} \rangle) \mid (\langle \text{ЛВ} \rangle \wedge \langle \text{ЛВ} \rangle) \mid (\langle \text{ЛВ} \rangle \vee \langle \text{ЛВ} \rangle)$

Вычислить (как boolean) значение этого выражения. Знаки \neg , \wedge , \vee означают соответственно отрицание, конъюнкцию И ДИЗЬЮНКЦИЮ.

6. Используя рекурсию, подсчитать сумму элементов из всех листьев дерева.

7. Найти минимальное подмножество вершин заданного орграфа, от которых достижимы все остальные его вершины.

Практические работы по курсу «Методы программирования»

Вариант 4

Задание: Разработать алгоритмы для решения задач.

2. Даны две действительные квадратные матрицы порядка n . Получить новую матрицу:

а) умножением элементов каждой строки первой матрицы на наибольшее из значений элементов соответствующей строки второй матрицы;

б) прибавлением к элементам каждого столбца первой матрицы произведения элементов соответствующих строк второй матрицы.

3. Список участков предприятия с разбивкой по цехам. Количество цехов и участков каждого цеха задать самостоятельно. Составить модуль поиска участка с максимальным номером.

5. Используя стек, решить следующую задачу.

В текстовом файле T записан текст, сбалансированный по круглым скобкам:

$\langle \text{текст} \rangle ::= \langle \text{пусто} \rangle \mid \langle \text{элемент} \rangle \langle \text{текст} \rangle$

$\langle \text{элемент} \rangle ::= \langle \text{буква} \rangle \mid (\langle \text{текст} \rangle)$

Требуется для каждой пары скобок напечатать номера их позиций в тексте в порядке возрастания номеров позиций открывающих скобок. Например, для текста $A + (45 - F(X) * (B - C))$ нужно напечатать

3 17; 8 10; 12 16.

6. Разработать программу, формирующую динамическую структуру данных для хранения генеалогического дерева. Каждая вершина дерева должна содержать следующую информацию: имя и год рождения.

7. Определить, изоморфны ли два заданных графа.

Практические работы по курсу
«Методы программирования»

Вариант 5

Задание: Разработать алгоритмы для решения задач.

2. Совокупность следующих друг за другом равных по значению элементов целочисленного массива назовем подпоследовательностью. Найти подпоследовательность максимальной длины.

3. Список работников цеха с разбивкой по профессиям. Количество профессий и работников каждой профессии цеха задано a_{ij} постоянно. Составить модуль сортировки фамилий по алфавиту.

5. Используя очередь, решить следующую задачу.

TYPE FR - FILE OF REAL;

За один просмотр файла f типа FR и без использования дополнительных файлов напечатать элементы файла f в следующем порядке: сначала -- все числа, меньшие a , затем -- все числа из отрезка $[a, b]$, и наконец - все остальные числа, сохраняя исходный взаимный порядок в каждой из этих групп чисел (a, b - заданы, $a < b$).

6. Используя рекурсию, найти самое большое число на n -ом уровне непустого дерева T (корень считать вершиной нулевого уровня).

7. Определить, является ли заданный граф двудольным.

Практические работы по курсу
«Методы программирования»

Вариант 6

Задание: Разработать алгоритмы для решения задач.

2. Дана действительная квадратная матрица порядка n . Преобразовать матрицу по правилу: строку с номером n сделать столбцом с номером n , а столбец с номером n сделать строкой с номером n .

3. Состав парка ЭВМ вычислительного центра с разбивкой ЭВМ по сериям. Количество серий и ЭВМ разных серий задано самостоятельно. Составить модуль сортировки ЭВМ по возрастанию объема оперативной памяти.

5. Используя стек, решить следующую задачу. В текстовом файле записана без ошибок формула следующего вида:

$\langle \text{формула} \rangle ::= \langle \text{цифра} \rangle \mid M(\langle \text{формула} \rangle, \langle \text{формула} \rangle) \mid m(\langle \text{формула} \rangle, \langle \text{формула} \rangle)$

$\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$, где M обозначает функцию \max , а m - \min .

Вычислить как целое число значение данной формулы. Например, $M(5, m(6, 8)) = 6$.

6. Разработать булеву функцию, которая определяет, равны ли два бинарных дерева.

7. По заданному графу G построить граф его транзитивного замыкания, т.е. такой граф G' , в качестве вершин которого берется множество вершин из G . Две вершины u, v в G' смежны тогда и только тогда, когда в G существует путь из u в v .

Практические работы по курсу
«Методы программирования»

Вариант 7

Задание: Разработать алгоритмы для решения задач.

2. Таблица футбольного чемпионата, в котором участвовало n команд, задана своей верхней правой частью в виде последовательности чисел 0, 1 или 2: первые $n-1$ чисел последовательности относятся к первой строке таблицы, следующие $n-2$ чисел – ко второй и т.д. Построить таблицу целиком, т.е. получить соответствующую квадратную матрицу порядка n (элементы главной диагонали заполняются нулями).

3. Список номеров и маршрутов рейсов автобусов с разбивкой рейсов по районам следования. Количество районов и рейсов в каждый район задать самостоятельно. Составить модуль сортировки рейсов по убыванию номеров.

5. Описать функцию `value (postfix)`, которая вычисляет как целое число значение выражения (без переменных), записанного в постфиксной форме в текстовом файле `postfix`.

Использовать следующий алгоритм вычисления. Выражение просматривается слева направо. Если встречается операнд (число), то его значение (как целое) заносится в стек, а если встречается знак операции, то из стека извлекаются два последних элемента, над ними выполняется операция и её результат записывается в стек. В конце концов в стеке остается только одно число - значение выражения.

6. Используя рекурсию, подсчитать число вершин на n -ом уровне непустого дерева T (корень считать вершиной нулевого уровня).

7. Определить число компонент связности графа.

Практические работы по курсу
«Методы программирования»

Вариант 8

Задание: Разработать алгоритмы для решения задач.

2. Даны натуральное число n , символы s_1, \dots, s_n .

а) Подсчитать наибольшее количество идущих подряд пробелов.

б) Выяснить, верно ли, что в последовательности s_1, \dots, s_n имеются пять идущих подряд букв `e`.

3. Список ПО на лазерных дисках в вычислительном центре с разбивкой по операционным системам. Количество ОС и дисков для каждой системы задать самостоятельно. Составить модуль поиска элементов списка по его информационным полям.

5. Разработать программу формирования стека, куда помещаются целые положительные числа, вводимые с клавиатуры. Процесс ввода должен прекращаться, как только среди вводимых чисел появляется отрицательное число. После этого программа должна вывести на экран монитора содержимое стека, при этом порядок выводимых чисел должен быть обратным по сравнению с последовательностью их ввода.

6. Используя рекурсию, определить максимальную глубину непустого дерева T , т.е. число ветвей в самом длинном из путей от корня дерева до листьев.

7. Найти минимальное покрывающее дерево графа.

Практические работы по курсу
«Методы программирования»

Вариант 9

Задание: Разработать алгоритмы для решения задач.

2. Дана целочисленная матрица размера 6×9 . Найти матрицу, получающуюся из данной:

- а) перестановкой столбцов – первого с последним, второго с предпоследним и т.д.;
- б) перестановкой строк – первой с последней, второй – с предпоследней и т.д.

3. Состав спортивного соревнования с разбивкой по видам спорта. Количество видов спорта и участников каждого вида спорта задать самостоятельно. Составить модуль формирования нового списка спортсменов, содержащего фамилии каждого второго спортсмена.

5. Организовать очередь из n целых чисел. Изменить ссылки так, чтобы последний элемент очереди стал первым, первый - вторым, второй - третьим и т.д.

6. Используя рекурсию, напечатать элементы из всех листьев дерева T .

7. Построить эйлеров цикл в эйлеровом графе

Практические работы по курсу
«Методы программирования»

Вариант 10

Задание: Разработать алгоритмы для решения задач.

2. Задан целочисленный массив A длиной n . Убывающей

"пилой" назовем такую последовательность $A[i], \dots, A[j]$, что для каждой тройки элементов последовательности $A[k], A[k+1], A[k+2]$ выполняется одно из условий:

- 1) $A[k] < A[k+1] > A[k+2]$ и $A[k] > A[k+2]$;
- 2) $A[k] > A[k+1] < A[k+2]$ и $A[k] > A[k+2]$.

Глубиной "пилы" назовем разность между максимальным и минимальным элементами массива в ней. Найти самую глубокую "пилу". Трудоемкость алгоритма должна быть линейной от n .

3. Программа работы конференции с разбивкой докладов по секциям. Количество секций и докладов в каждой секции задать самостоятельно. Составить модуль формирования нового списка, содержащего доклады с несколькими авторами.

5. Разработать программу формирования очереди, содержащей целые числа, и упорядочивания по возрастанию элементов в этой очереди. В процессе упорядочивания элементы очереди перемещаться не должны.

6. Описать рекурсивную функцию или процедуру, находящую величину наибольшего элемента дерева T .

7. Решить задачу коммивояжера.

Практические работы по курсу
«Методы программирования»

Вариант 11

Задание: Разработать алгоритмы для решения задач.

2. Задана ломаная линия набором n точек плоскости в виде массивов координат $\{X[1], Y[1]\}, \dots, \{X[n], Y[n]\}$. Часть ломаной с i -й точки по j -ю, для которой выполнено условие:

$$\{X[i] \leq X[i+1], Y[i] \leq Y[i+1]\}, \dots, \\ \{X[j-1] \leq X[j], Y[j-1] \leq Y[j]\}$$

назовем восходящей трассой, а евклидово расстояние между i -й и j -й точками — длиной восходящей трассы. (Расстояние между точками равно квадратному корню из суммы квадратов разностей координат). Вывести точки линии по трассам в порядке убывания длин трасс.

3. Конструкция технического устройства, состоящего из нескольких блоков. Причем каждый блок содержит определенное количество модулей. Составить модуль формирования нового списка, из которого удаляется модуль с заданным наименованием.

5. Используя стек, решить следующую задачу. Проверить, является ли содержимое текстового файла T правильной записью формулы следующего вида:

$$\langle \text{формула} \rangle ::= \langle \text{терм} \rangle | \langle \text{терм} \rangle + \langle \text{формула} \rangle \langle \text{терм} \rangle - \langle \text{формула} \rangle \\ \langle \text{терм} \rangle ::= \langle \text{имя} \rangle | (\langle \text{формула} \rangle) | [\langle \text{формула} \rangle] | \{\langle \text{формула} \rangle\} \\ \langle \text{имя} \rangle ::= x | y | z .$$

6. Разработать рекурсивную функцию, которая определяет, помещена ли строка в бинарное дерево поиска.

7. Найти все мосты заданного графа.

Практические работы по курсу
«Методы программирования»

Вариант 12

Задание: Разработать алгоритмы для решения задач.

2. Задана ломаная линия набором n точек плоскости в виде массивов координат $\{X[1], Y[1]\}, \dots, \{X[n], Y[n]\}$. Часть ломаной с i -й точки по j -ю, для которой выполнено условие:

$$\{X[i] \leq X[i+1], Y[i] \geq Y[i+1]\}, \dots, \\ \{X[j-1] \leq X[j], Y[j-1] \geq Y[j]\} \text{ назовем нисходящей трассой, при условии:} \\ \{X[i] \leq X[i+1], Y[i] \leq Y[i+1]\}, \dots, \\ \{X[j-1] \leq X[j], Y[j-1] \leq Y[j]\}$$

— восходящей трассой, а количество точек трассы — мощностью трассы. Вывести точки линии по трассам в порядке возрастания мощностей трасс, вначале для нисходящих трасс, а затем для восходящих трасс.

Определить трудоемкость алгоритма в зависимости от n и m , учитывая, что $m \ll n$, где m — количество трасс.

3. Конструкция модуля, состоящая из микросхем. Количество микросхем и вводов-выводов каждой микросхемы задать самостоятельно. Составить модуль формирования нового списка вводов-выводов, из которого удаляются элементы с номерами 7 и 14.

5. Последовательность символов, ограниченную точкой, занести в два стека, содержащих гласные и согласные буквы. Вывести текст и элементы из обоих стеков.

6. Описать рекурсивную процедуру или функцию, определяющую число вхождений элемента E в дерево T .

7. Найти минимальное подмножество ребер, удаление которых превращает заданный граф в несвязный.

Практические работы по курсу

«Методы программирования»

Вариант 13

Задание: Разработать алгоритмы для решения задач.

2. Задан целочисленный массив A из n элементов. Кольцевой цепочкой назовем такие элементы $A[i_1]$, $A[i_2]$, ..., $A[i_m]$, что:
 $A[i_1]=i_2$, $A[i_2]=i_3$, ..., $A[i_m]=i_1$.
Найти самую длинную кольцевую цепочку в массиве A .
3. Конструкция вычислительной сети, состоящей из узлов разных типов. Количество узлов и типов узлов задать самостоятельно. Составить модуль формирования нового списка узлов, в котором добавлен новый заданный узел.
5. Последовательность символов, ограниченную точкой, занести в две очереди, содержащие гласные и согласные буквы русского алфавита. Вывести элементы очереди гласных и согласных букв.
6. Подсчитать число вершин на n -ом уровне непустого дерева T (корень считать вершиной нулевого уровня).
7. Построить гамильтонов путь в графе.

Практические работы по курсу
«Методы программирования»

Вариант 14

Задание: Разработать алгоритмы для решения задач.

2. Дан упорядоченный целочисленный массив. Сформировать второй массив из всех таких различных значений, которые в первом массиве встречаются по два и более раза. Алгоритм должен иметь линейную трудоемкость.
3. Книга состоит из глав и параграфов. Количество глав и параграфов каждой главы задать самостоятельно. Составить модуль формирования нового списка параграфов, в котором первый и последний элементы поменялись местами.
5. Коэффициенты многочлена
 $Q(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0 x^0$ занести в список.
Преобразовать элементы списка так, чтобы они содержали коэффициенты первой производной многочлена.
6. Вычислить среднее арифметическое всех элементов непустого двоичного дерева T .
7. Найти обратный граф заданному.

Практические работы по курсу
«Методы программирования»

Вариант 15

Задание: Разработать алгоритмы для решения задач.

2. Даны два упорядоченных целочисленных массива: A , содержащий n_1 элементов, и B из n_2 элементов. Переписать из массива A в массив C все такие элементы, значения которых не совпадают ни с одним из значений элементов массива B . При этом требуется, чтобы массив C был упорядоченным. Алгоритм должен иметь линейную трудоемкость.

3. Задачники состоят из списка задач, разбитых на темы. Количество тем и задач задать самостоятельно. Составить модуль формирования нового списка задач, в котором после задачи с заданным номером записан элемент с новой задачей.

5. Для очереди, представленной в виде массива, склеенного в кольцо (так, что если очередь достигает правого края массива, то новые элементы записываются в начало массива), описать следующие процедуры или функции:

создание пустой очереди Q (очистка очереди);

проверка, является ли очередь Q пустой;

добавление в конец очереди Q элемента x ($inoch(Q,x)$);

удаление из очереди Q первого элемента с присвоением его значения параметру x ($outoch(Q,x)$). Если операция по каким-либо причинам не может быть выполнена, следует вызывать некоторую процедуру $Erro(k)$, где k - номер ошибки: 1- переполнение очереди, 2-исчерпание очереди.

6. Используя очередь или стек, описать процедуру или функцию, определяющую число вхождений элемента E в дерево T .

7. Определить является ли заданный граф трехдольным.

Практические работы по курсу
«Методы программирования»

Вариант 16

Задание: Разработать алгоритмы для решения задач.

2. Задан целочисленный массив. Подсчитать число различных значений элементов в массиве.

3. Набор материалов, имеющихся на складе с разбивкой по виду товара. Количество видов товаров и штук каждого товара задать самостоятельно. Составить модуль формирования двух новых списков из исходного списка товаров. В первый список попадают номера товаров меньше некоторого заданного, во второй - больше.

5. Для стека, представленного в виде массива из n компонент, описать следующие процедуры или функции:

создание пустого стека S (очистка стека);

проверка, является ли стек S пустым;

добавление в стек S элемента x ($instack(Q,x)$);

удаление из стека S элемента с присвоением его значения параметру x ($outstack(QX)$)

Если операция по каким-либо причинам не может быть выполнена, следует вызывать некоторую процедуру $Erro(k)$, где k - номер ошибки: 1- переполнение стека, 2- исчерпание стека.

6. Используя очередь или стек, описать процедуру или функцию, которая ищет элемент самого левого листа непустого дерева T .

7. Определить является ли заданный граф планарным.

Практические работы по курсу
«Методы программирования»

Вариант 17

Задание: Разработать алгоритмы для решения задач.

2. Даны действительная матрица размера $n \times (n+1)$, действительные числа $a_1, \dots, a_{n+1}, b_1, \dots, b_{n+1}$, натуральные числа p, q ($p \leq n, q \leq n+1$). Образовать новую матрицу размера $(n+1) \times (n+2)$ вставкой после строки с номером p данной матрицы новой строки с элементами a_1, \dots, a_{n+1} и последующей вставкой после столбца с номером q нового столбца с элементами b_1, \dots, b_{n+1} .

3. Чтобы зашифровать текст, записанный с помощью русских букв и знаков препинания, его можно переписать, заменив каждую букву непосредственно следующей за ней по алфавиту (буква я заменяется на а). Можно обобщить способ шифровки, изложенный в предыдущей задаче – сдвиг производится не на одну букву, а на n букв, где n – данное натуральное число (можно представлять себе, что буквы выписаны по кругу, как цифры на циферблате).

а) Зашифровать данный текст.

б) Расшифровать данный текст.

4. Используя очередь, решить следующую задачу. Содержимое текстового файла f , разделенное на строки, переписать в текстовый файл g , перенося при этом в конец каждой строки все входящие в нее цифры (с сохранением исходного порядка как среди цифр, так и среди остальных литер строки).

5. Используя рекурсию, найти самое большое число на n -ом уровне непустого дерева T (корень считать вершиной нулевого уровня).

6. Построить гамильтонов путь в графе

Практические работы по курсу
«Методы программирования»

Вариант 18

Задание: Разработать алгоритмы для решения задач.

2. Дан целочисленный массив. Найти в нем подпоследовательность максимальной длины, представляющую собой арифметическую прогрессию.

3. Один из простейших способов шифровки текста состоит в табличной замене каждого символа другим символом – его шифром. Выбрать некоторую таблицу, разработать способ ее представления, затем

а) зашифровать данный текст;

б) расшифровать данный текст.

4. Используя очередь, решить следующую задачу.

TYPE имя = (Анна, ..., Яков);

дети = ARRAY[имя, имя] OF BOOLEAN; потомки = FILE OF имя;

Считая заданным имя I и массив D типа дети ($D[X, Y] = \text{TRUE}$, если человек по имени Y является ребенком человека по имени X), записать в файл P типа потомки имена всех потомков человека с именем I в следующем порядке: сначала - имена всех его детей, затем - всех его внуков, затем - правнуков и т.д.

5. Используя рекурсию, подсчитать сумму элементов из всех листьев дерева.

6. Найти минимальное подмножество ребер, удаление которых превращает заданный граф в несвязный

Практические работы по курсу
«Методы программирования»

Вариант 19

Задание: Разработать алгоритмы для решения задач.

2. Даны целые числа a_1, \dots, a_{10} , целочисленная квадратная матрица порядка n . Заменить нулями в матрице те элементы с четной суммой индексов, для которых имеются равные среди a_1, \dots, a_{10} .

3. Чтобы зашифровать текст из 121 буквы, его можно записать в квадратную матрицу порядка 11 по строкам, а затем прочитать по спирали, начиная с центра (т.е. с элемента, имеющего индексы 6, 6).

а) Зашифровать данный текст.

б) Расшифровать данный текст.

4. Используя стек, решить следующую задачу.

В текстовом файле LOG записано без ошибок логическое выражение (ЛВ) в следующей форме:

$\langle \text{ЛВ} \rangle ::= \text{true} \mid \text{false} \mid (\neg \langle \text{ЛВ} \rangle) \mid (\langle \text{ЛВ} \rangle \wedge \langle \text{ЛВ} \rangle) \mid (\langle \text{ЛВ} \rangle \vee \langle \text{ЛВ} \rangle)$

Вычислить (как boolean) значение этого выражения. Знаки \neg , \wedge , \vee означают соответственно отрицание, конъюнкцию И ДИЗЬЮНКЦИЮ.

5. Используя рекурсию, подсчитать число вершин на n -ом уровне непустого дерева T (корень считать вершиной нулевого уровня).

6. Найти минимальное покрывающее дерево графа.

Практические работы по курсу
«Методы программирования»

Вариант 20

Задание: Разработать алгоритмы для решения задач.

2. Задан числовой массив A из n элементов. Ступенькой назовем такие элементы $A[i]$, $A[i+1]$, ..., $A[j]$, что:

1) $A[k] < A[k+1]$, $k=i, i+1, \dots, j-1$;

2) $A[i-1] \geq A[i]$, если $i > 1$ и $A[j] \geq A[j+1]$, если $j < n$. Высотой ступеньки назовем разность $A[j] - A[i]$, а длиной - количество элементов массива, входящих в ступеньку. Вычислить:

1) высоту и длину каждой из ступенек;

2) высоту самой длинной ступеньки;

3) длину самой высокой ступеньки.

3. Шифровка текста с помощью решетки заключается в следующем. Решетка, т.е. квадрат из клетчатой бумаги 10×10 клеток, некоторые клетки в котором вырезаны, совмещается с целым квадратом 10×10 клеток и через прорезы на бумагу наносятся первые буквы текста. Затем решетка поворачивается на 90° и через прорезы записываются следующие буквы. Это повторяется еще дважды. Таким образом, на бумагу наносится 100 букв текста. Решетку можно изображать квадратной матрицей порядка 10 из нулей и единиц (нуль изображает прорезь). Доказать, что матрица $[a_{ij}]_{i=1, \dots, 10}$ может служить ключом шифра, если из элементов a_{ij} , $a_{10-i+ij}$, $a_{i10-j+1}$, $a_{10-i+1; 10-j+1}$ в точности один равен нулю.

Даны последовательность из 100 букв и матрица-ключ.

а) Зашифровать данную последовательность.

б) Расшифровать данную последовательность.

4. Используя очередь, решить следующую задачу.

TYPE FR - FILE OF REAL;

За один просмотр файла f типа FR и без использования дополнительных файлов напечатать элементы файла f в следующем порядке: сначала -- все числа, меньшие a , затем - все числа из отрезка $[a, b]$, и наконец - все остальные числа, сохраняя исходный взаимный порядок в каждой из этих групп чисел (a, b - заданы, $a < b$).

5. Используя стек, решить следующую задачу. В текстовом файле записана без ошибок формула следующего вида:

$\langle \text{формула} \rangle ::= \langle \text{цифра} \rangle \mid M(\langle \text{формула} \rangle, \langle \text{формула} \rangle) \mid m(\langle \text{формула} \rangle, \langle \text{формула} \rangle)$

$\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$, где M обозначает функцию \max , а m - \min .

Вычислить как целое число значение данной формулы. Например, $M(5, m(6, 8)) = 6$.

6. Определить число компонент связности графа.

Практические работы по курсу «Методы программирования»

Вариант 21

Задание: Разработать алгоритмы для решения задач.

2. Даны натуральное число n , символы s_1, \dots, s_n . Преобразовать последовательность s_1, \dots, s_n , заменив в ней:

- все восклицательные знаки точками;
- каждую точку многоточием (т.е. тремя точками);
- каждую из групп стоящих рядом точек одной точкой;
- каждую из групп стоящих рядом точек многоточием (т.е. тремя точками).

3. Зафиксируем натуральное k и перестановку чисел $1, \dots, k$ (ее можно задать с помощью последовательности натуральных чисел p_1, \dots, p_k , в которую входит каждое из чисел $1, \dots, k$). При шифровке в исходном тексте k каждой из последовательных групп по k символов применяется зафиксированная перестановка. Пусть $k = 4$ и перестановка есть $3, 2, 4, 1$. Тогда группа символов s_1, s_2, s_3, s_4 заменяется на s_3, s_2, s_4, s_1 . Если в последней группе меньше четырех символов, то к ней добавляются пробелы. Пользуясь изложенным способом:

- зашифровать данный текст;
- расшифровать данный текст.

4. Разработать программу формирования стека, куда помещаются целые положительные числа, вводимые с клавиатуры. Процесс ввода должен прекращаться, как только среди вводимых чисел появляется отрицательное число. После этого программа должна вывести на экран монитора содержимое стека, при этом порядок выводимых чисел должен быть обратным по сравнению с последовательностью их ввода.

5. Вычислить среднее арифметическое всех элементов непустого двоичного дерева T .

6. Решить задачу раскраски графа. (полный перебор)

Практические работы по курсу «Методы программирования»

Вариант 22

Задание: Разработать алгоритмы для решения задач.

2. Дан целочисленный массив A длиной n . Назовем элемент $A[i]$ основанием ступеньки, если выполняется соотношение $A[i] < A[i+1] < \dots < A[i+k]$. Величина k - высота ступеньки. Найти в массиве основания ступенек и высоту каждой из них.

3. Следующий способ предназначен для шифровки последовательностей нулей и единиц (или же, например, точек и тире). Пусть a_1, \dots, a_n – такая последовательность. То, что предлагается в качестве ее шифра – это последовательность b_1, \dots, b_n , образованная по следующему закону:

$$b_1 = a_1, \quad b_i = \begin{cases} 1, & \text{если } a_i = a_{i-1}, \\ 0 & \text{в противном случае} \end{cases} \quad (i = 2, \dots, n).$$

Пользуясь изложенным способом:

а) Зашифровать данную последовательность.

б) Расшифровать данную последовательность.

4. Используя стек, решить следующую задачу.

В текстовом файле T записан текст, сбалансированный по круглым скобкам:

$\langle \text{текст} \rangle ::= \langle \text{пусто} \rangle \mid \langle \text{элемент} \rangle \langle \text{текст} \rangle$

$\langle \text{элемент} \rangle ::= \langle \text{буква} \rangle \mid (\langle \text{текст} \rangle)$

Требуется для каждой пары скобок напечатать номера их позиций в тексте в порядке возрастания номеров позиций открывающих скобок. Например, для текста $A + (45 - F(X) * (B - C))$ нужно напечатать

3 17; 8 10; 12 16.

5. Используя очередь или стек, описать процедуру или функцию, определяющую число вхождений элемента E в дерево T .

6. Найти кратчайший путь из одной вершины в другую (поиск в ширину).

Практические работы по курсу
«Методы программирования»

Вариант 23

Задание: Разработать алгоритмы для решения задач.

2. Дана целочисленная матрица размера 6×9 . Найти матрицу, получающуюся из данной:

а) перестановкой столбцов – первого с последним, второго с предпоследним и т.д.;

б) перестановкой строк – первой с последней, второй – с предпоследней и т.д.

3. “Исправление ошибок”. Пусть по некоторому каналу связи передается сообщение, имеющее вид последовательности нулей и единиц (или, аналогично, точек и тире). Из-за помех возможен ошибочный прием некоторых сигналов: ноль может быть воспринят как единица и наоборот. Можно передавать каждый сигнал трижды, заменяя, например, последовательность 1, 0, 1 последовательностью 1,1,1,0,0,0,1,1,1. Три последовательные цифры при расшифровке заменяются той цифрой, которая встречается среди них по крайней мере дважды. Такое утраивание сигналов существенно повышает вероятность правильного приема сообщения.

Написать программу расшифровки.

4. Подсчитать число вершин на n -ом уровне непустого дерева T (корень считать вершиной нулевого уровня).

5. Организовать очередь из n целых чисел. Изменить ссылки так, чтобы последний элемент очереди стал первым, первый - вторым, второй - третьим и т.д.

6. Определить, изоморфны ли два заданных графа.

Практические работы по курсу
«Методы программирования»

Вариант 24

Задание: Разработать алгоритмы для решения задач.

2. Дана действительная квадратная матрица порядка n . Преобразовать матрицу по правилу: строку с номером n сделать столбцом с номером n , а столбец с номером n сделать строкой с номером n .

3. Список сотрудников предприятия с разбивкой по должностям. Количество должностей и количество сотрудников, имеющих одинаковую должность задать самостоятельно. Составить модуль формирования списка однофамильцев, работающих на одной должности.

4. Разработать программу формирования очереди, содержащей целые числа, и упорядочивания по возрастанию элементов в этой очереди. В процессе упорядочивания элементы очереди перемещаться не должны.

5. . Описать рекурсивную функцию или процедуру, находящую величину наибольшего элемента дерева T .

6. Определить, является ли заданный граф двудольным.

Практические работы по курсу
«Методы программирования»

Вариант 25

Задание: Разработать алгоритмы для решения задач.

2. Даны две действительные квадратные матрицы порядка n . Получить новую матрицу:

а) умножением элементов каждой строки первой матрицы на наибольшее из значений элементов соответствующей строки второй матрицы;

б) прибавлением к элементам каждого столбца первой матрицы произведения элементов соответствующих строк второй матрицы.

3. Список вузов страны с разбивкой по профилям. Количества профилей и вузов каждого профиля задать самостоятельно. Составить модуль формирования нового списка вузов заданного профиля, являющихся университетами.

4. Используя стек, решить следующую задачу. Проверить, является ли содержимое текстового файла T правильной записью формулы следующего вида:

$\langle \text{формула} \rangle ::= \langle \text{терм} \rangle \mid \langle \text{терм} \rangle + \langle \text{формула} \rangle \mid \langle \text{терм} \rangle - \langle \text{формула} \rangle$

$\langle \text{терм} \rangle ::= \langle \text{имя} \rangle \mid (\langle \text{формула} \rangle) \mid [\langle \text{формула} \rangle] \mid \{\langle \text{формула} \rangle\}$

$\langle \text{имя} \rangle ::= x \mid y \mid z$.

5. В непустом дереве T найти длину пути (число ветвей) от корня до ближайшей вершины с элементом E . Если элемент E не входит в T , то за ответ принять "-1".

6. Решить задачу раскраски графа.

Практические работы по курсу
«Методы программирования»

Вариант 26

Задание: Разработать алгоритмы для решения задач.

2. В данной действительности матрице размера $n \times m$ ($n \geq 3, m \geq 3$) поменять местами:

а) строки с номерами 2 и $n-1$;

б) столбцы с номерами 3 и $n-2$.

3. Список номеров автомобилей, паркующихся на платной стоянке с разбивкой по маркам. Количество марок и автомобилей каждой марки задать самостоятельно. Составить модуль формирования нового списка, содержащего номера с заданной частью номера.

4. Последовательность символов, ограниченную точкой, занести в две очереди, содержащие гласные и согласные буквы русского алфавита. Вывести элементы очереди гласных и согласных букв.

Используя рекурсию, подсчитать сумму элементов из всех листьев дерева.

Решить задачу коммивояжера.

Практические работы по курсу
«Методы программирования»

Вариант 27

Задание: Разработать алгоритмы для решения задач.

2. Задана последовательность натуральных чисел из диапазона $[1, 2147483647]$. Количество чисел в этой последовательности не превышает 100000. Необходимо определить, можно ли выстроить эти числа в отрезок арифметической прогрессии. При необходимости для построения прогрессии порядок чисел в последовательности можно изменять.

Например, из чисел последовательности

9 6 15 21 12 18

МОЖНО построить арифметическую прогрессию, а из чисел последовательности 12456789 НЕЛЬЗЯ построить прогрессию.

3. Способы шифровки текста, основанные на табличной замене входящих в него букв некоторыми символами (см. задачу 836) или на сдвиге букв (см. задачи 834, 835), нехороши тем, что шифр может быть разгадан путем частотного анализа символов, входящих в зашифрованный текст – вспомним рассказ Э.По “Золотой жук”. Следующий способ лишен этого недостатка. Пусть шифруемым текстом является последовательность, состоящая из букв s_0, s_1, \dots, s_n . С помощью датчика случайных чисел можно, например, получать члены последовательности v_0, v_1, \dots с распределением

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$$

и при шифровке символа s_i ($0 \leq i \leq n$) применять сдвиг на v_i букв. Тогда две встречающиеся в зашифрованном тексте одинаковые буквы не обязаны обозначать одну и ту же букву исходного текста. Для расшифровки сообщения нужно воспользоваться таким же датчиком случайных чисел.

В следующих двух заданиях для простоты предполагается, что текст записан русскими буквами без знаков препинания, пробел после шифровки остается пробелом.

а) Зашифровать данный текст.

б) Расшифровать данный текст.

4. Разработать программу, формирующую динамическую структуру данных для хранения генеалогического дерева. Каждая вершина дерева должна содержать следующую информацию: имя и год рождения.

5. Для очереди, представленной в виде массива, склеенного в кольцо (так, что если очередь достигает правого края массива, то новые элементы записываются в начало массива), описать следующие процедуры или функции:

создание пустой очереди Q (очистка очереди);

проверка, является ли очередь Q пустой;

добавление в конец очереди Q элемента x ($inoch(Q,x)$);

удаление из очереди Q первого элемента с присвоением его значения параметру x ($outoch(Q,x)$). Если операция по каким-либо причинам не может быть выполнена, следует вызывать некоторую процедуру $Error(k)$, где k - номер ошибки: 1- переполнение очереди, 2-исчерпание очереди.

6. Найти минимальное подмножество ребер, удаление которых превращает заданный граф в несвязный

ПРИЛОЖЕНИЕ 2 Требования к оформлению отчетов

Отчет по лабораторной или контрольной работе должен содержать следующие обязательные пункты:

- титульный лист
- цель работы
- задание
- краткую теорию и ход выполнения работы:
- основные теоретические положения
- спецификации разрабатываемой программы
- проект программы
- результаты выполнения работы, тест, доказывающий правильность работы программы
- выводы
- приложение – исходный текст программы с комментариями

Пример титульного листа:

<p style="text-align: center;">МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР) Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)</p> <p style="text-align: center;">ОТЧЕТ По № "N" "<наименование работы>"</p> <p style="text-align: right;">Выполнил: Студент гр. 520 Ф.И.О. Принял: Ф.И.О преподавателя</p> <p style="text-align: center;">2012</p>

Спецификации:

- тип используемых данных и диапазон их изменения на входе;
- интерфейс и сообщения системы;
- особые ситуации;
- выходные данные;
- алгоритмы обработки данных.

Проект программы:

Общая структура, исходя из спецификаций (модули, функции, объекты, классы).

Результаты выполнения программы:

А) Требования к листингу программы.

Записывается один оператор на строке.

Целочисленные переменные, используемые для организации циклов, обозначаются - i, j, k, l, m, n.

Комментарии обязательны. (заголовочные - указывается назначение программы и автор, описываются основные переменные и процедуры. Текущие – по ходу программы.)

Отступы обязательны.

Б) Должен быть задан тест и результат выполнения теста. (проверены все критические ситуации.)

В) Необходимо оценить трудоемкость алгоритма.

Выводы

Выводы по работе обязательны.