

Содержание

1	Практическая работа №1. Работа с опциями транслятора(4 часа)	1
2	Практическая работа №2. Трансляция программ, состоящих из нескольких объектных файлов (2 часа)	1
3	Практическая работа №3. Трансляция программ, состоящих из нескольких объектных файлов на различных языках программирования (2 часа)	2
4	Практическая работа №4. Средства контроля версий (2 часа)	3
5	Практическая работа №5. Создание файлов помощи (2 часа)	3
6	Практическая работа №6. Профилирование (2 часа)	3
7	Практическая работа №7. Поиск утечек памяти (2 часа)	4
8	Практическая работа №8. Автоматизация тестирования GUI (2 часа)	5
9	Практическая работа №9. Автоматизация тестирования WEB-интерфейса (2 часа)	5
10	Практическая работа №10. UNIT-тестирование (2 часа)	5
11	Практическая работа №11. Статический анализ кода (2 часа)	5
12	Практическая работа №12. Средства управления требованиями (2 часа)	5
13	Практическая работа №13. Системы отслеживания ошибок (2 часа)	5
14	Практическая работа №14. Средство создания документации (4 часа)	6
15	Практическая работа №15. Инструментальные средства проектирования (4 часа)	6
16	Практическая работа №16. Средства обратной разработки (2 часа)	7
17	Практическая работа №17. Средства автоматизированного проектирования (4 часа)	8

1	Практическая работа №1. Работа с опциями транслятора(4 часа)
2	Практическая работа №2. Трансляция программ, состоящих из нескольких объектных файлов (2 часа)

Напишите файловую базу данных, управляемую в консольном интерактивном режиме (квазиоконный интерфейс делать не нужно) с возможностями: добавление, удаление, изменение информации соответственно вашему варианту (на языке C). В процессе разработки осуществите разбиение программы на отдельные объектные файлы, а также подготовьте Makefile для автоматизации сборки.

Сборку лучше осуществлять с помощью gcc

- п2-0-1 База данных адресов и телефонов. Поля: Фамилия, Имя, Отчество, Адрес, Телефон.
- п2-0-2 База данных учеников. Поля: Фамилия, Имя, Номер группы, Оценка по информатике.
- п2-0-3 База данных маршрутов автобусов. Поля: номер маршрута, первая остановка, последняя остановка, время поездки по всему маршруту.
- п2-0-4 База данных книг. Поля: Автор, Название, Количество страниц, Тематика.
- п2-0-5 База данных государств. Поля: название государства, его территория (площадь), его население (количество).
- п2-0-6 База данных газет. Поля: название, периодичность выхода, количество страниц, цена.
- п2-0-7 База данных пейджеров. Поля: название, количество строк, стоимость.
- п2-0-8 База данных библиотек. Поля: номер библиотеки, адрес библиотеки, телефон библиотеки.
- п2-0-9 База данных жилищных субсидий. Поля: ФИО, средний доход в семье, размер дотации.
- п2-0-10 База данных институтов. Поля: название, конкурс (число - количество претендентов на место при поступлении в институт), адрес, телефон.

- п2-0-11 База данных политиков. Поля: фамилия, имя, рейтинг, должность.
- п2-0-12 База данных домов. Поля: улица, номер дома, количество этажей.
- п2-0-13 База данных языков программирования. Поля: название, год создания, сложность (число от 1 до 5).
- п2-0-14 База данных школ. Поля: номер школы, адрес, телефон, комментарий.
- п2-0-15 База данных музеев. Поля: название музея, адрес, плата за вход.
- п2-0-16 База данных праздников. Поля: месяц и число, описание праздника, признак выходного дня.
- п2-0-17 База данных изделий завода. Поля: название изделия, цена изделия, вес изделия.
- п2-0-18 База данных деревьев в городе. Поля: название дерева, количество деревьев этого вида в городе, количество посадок в этом году такого вида деревьев.
- п2-0-19 База данных статей в номере журнала. Поля: название статьи, автор статьи, номер страницы.
- п2-0-20 База данных звёзд. Поля: название звезды, расположение звезды на небе (два числа: первое от 0 до 360, второе – от 0 до 180), яркость звезды (число от 1 до 5).
- п2-0-22 База данных улиц города. Поля: название улицы, количество домов на улице, протяжённость в километрах.
- п2-0-23 База данных музыкального коллектива. Поля: ФИО, название инструмента, зарплата.
- п2-0-24 База данных вокзалов страны. Поля: название вокзала, город, подчинение (например, «октябрьская железная дорога»).
- п2-0-25 База данных домашних собак города. Поля: кличка, порода, хозяин, адрес.

3 Практическая работа №3. Трансляция программ, состоящих из нескольких объектных файлов на различных языках программирования (2 часа)

Теоретическое введение

При вызове подпрограмм используются различные соглашения о способе их вызова. Транслятор gcc по умолчанию использует следующее соглашение для функций, имеющих три аргумента целого типа (ABI SysV):

- при использовании 32-битного режима для передачи параметров используется стек: в него записываются аргументы в обратном порядке (на каждый аргумент по 4 байта), после чего осуществляется вызов функции (при котором в стек заносится адрес возврата, занимающий также 4 байта). Таким образом первый аргумент расположен по адресу $[4+ESP]$, второй – $[8+ESP]$ и так далее. Возвращаемое значение передается через регистр EAX.
- при использовании 64-битного режима для передачи параметров используются регистры процессора: rdi, rsi, rdx; результат возвращается через регистр rax.

Для простоты в ваших программах рекомендуется сохранять значения всех регистров, кроме eax (rax).

Для трансляции объектного файла в исходном коде FASM записывается строка: `FORMAT ELF (FORMAT ELF64)` для 32-битного и 64-битного режима соответственно. Функция задается меткой, при этом необходимо написать строчку

```
public <название функции>
```

где название функции указывается без угловых скобок. В программе на языке Си указывается объявление вашей функции, описанной на языке Ассемблера.

Линковку следует осуществлять с помощью команды

```
gcc <список объектных файлов> -o <исполняемый файл>
```

Задание

Напишите программу, состоящую из двух объектных файлов: первый – написан на языке C (gcc), второй – на языке ассемблера (FASM). При этом программа на языке C должна осуществлять ввод трех целых чисел – x, y и z; а объектный файл, созданный на языке Ассемблера, должен вычислять значение выражения согласно вашему варианту.

Подготовьте Makefile для автоматизации сборки.

1. $(x+2*y+3*z)/4$
2. $(3*x+z*y)\%4$
3. $(x-y*z+3)/5$

4. $x/5+y*2-3*z$
5. $(x-2*y)/z$
6. $(2*x+z)\%y$
7. $x\%2-y/z$
8. $x*x+z\%y$
9. $(x/y+z)\%x$
10. $((10-x)*(y+2)\%z$
11. $(x*y-z\%5)/2$
12. $(x\%5+y|z)*3$
13. $(x-y\&(z+15))\%10$
14. $(3-x\%5)/(y+2*z)$
15. $(x+(y<<2)\%5+z)$
16. $(x/3+(y>>3)*z)$
17. $(x|3+y\&(\sim 3))\%125$
18. $((x*y*y-z)\%65)$
19. $(x/4+y\%z)<<2$
20. $(x\%5+z/y)>>2$
21. $(x|(y\&z))\%125$
22. $(x<<2+y>>2+z\%3)$
23. $(x/(y/z)+y*3)$
24. $(x|(\sim y))/z-23$
25. $(y\&(x|5)-z\%256)$

4 Практическая работа №4. Средства контроля версий (2 часа)

Осуществите создание репозитория (хотя бы локального) git и выполнение всех следующих работ осуществляясь с постоянной синхронизацией рабочей копии с репозиторием.

5 Практическая работа №5. Создание файлов помощи (2 часа)

Для любой из собственных курсовых работ разработайте файл помощи в формате СНМ с использованием средства HTMLHELP.

6 Практическая работа №6. Профилирование (2 часа)

Для программы, разработанной в рамках практической работы №2, проведите профилирование с помощью valgrind. Результаты профилирования представьте в отчете и осуществите его анализ в выводах. В случае необходимости произведите модификацию разработанной программы.

7 Практическая работа №7. Поиск утечек памяти (2 часа)

Для программы, описанной ниже, проведите поиск утечек памяти с помощью valgrind. Доведите вашу программу до такого состояния, чтобы утечек не было. Язык разработки: С или С++.

В программе во время обработки список должен просматриваться один раз. Программа состоит из трех частей: формирование списка (ввод с клавиатуры); обработка; вывод списка.

- 1 Нахождение суммы всех элементов исходного списка, удаление всех элементов, равных 0, добавление квадрата элемента после каждого элемента списка, полученного после завершения предыдущих преобразований.
- 2 Нахождение произведения всех элементов исходного списка, удаление всех элементов, меньших 0, добавление единицы после каждого четного элемента списка, полученного после завершения предыдущих преобразований.
- 3 Нахождение максимума всех элементов исходного списка, удаление всех четных элементов, добавление копии элемента после каждого элемента списка, пропорционального 3.
- 4 Нахождение минимума всех элементов исходного списка, удаление всех нечетных элементов, добавление после каждого элемента исходного списка его номера (каким он был в исходном списке).
- 5 Нахождение суммы всех четных элементов исходного списка, удаление всех чисел, оканчивающихся на 0, из списка, добавление после каждого элемента списка квадрата этого элемента.
- 6 Нахождение максимума всех четных элементов исходного списка, удаление всех нечетных элементов и добавление после каждого элемента списка, являющегося квадратом целого числа, значения -1 .
- 7 Нахождение минимума всех элементов, больших нуля; удаление всех отрицательных элементов и добавление после каждого элемента списка числа, на единицу большего.
- 8 Нахождение суммы всех элементов; удаление всех элементов, пропорциональных двум, добавление после каждого элемента списка куба этого элемента.
- 9 Нахождение суммы всех элементов, меньших нуля; удаление всех элементов, последняя цифра которых равна 5, добавление после каждого элемента списка числа, равного последней цифре данного элемента.
- 10 Нахождение наименьшего четного числа; удаление всех элементов, больших 10, добавление после каждого элемента списка квадрата данного числа.
- 11 Нахождение наименьшего числа, кратного трем; удаление всех элементов – двузначных натуральных чисел, добавление после каждого элемента, пропорционального двум, копии этого числа.
- 12 Нахождение наибольшего номера элемента списка (в исходном списке) числа, кратного трем; удаление всех элементов, в которых последние две цифры совпадают, добавление после каждого элемента числа, на единицу меньшего.
- 13 Нахождение суммы чисел, кратных трем; удаление всех элементов, оканчивающихся на 5, добавление после каждого положительного элемента, копии этого числа.
- 14 Нахождение произведения однозначных натуральных чисел; удаление всех элементов, меньших по модулю 5, добавление после каждого отрицательного элемента, копии этого числа.
- 15 Нахождение наибольшего номера элемента списка (в исходном списке) двузначного числа; удаление всех элементов, в которых последняя цифра больше 5, добавление после каждого элемента числа, на единицу большего.
- 16 Нахождение суммы всех чисел, кратных 5; удаление элементов, являющихся квадратом натурального числа, добавление после каждого положительного элемента его копии.
- 17 Нахождение произведения всех чисел, кратных 5; удаление элементов, стоящих на позиции, номер которой (в исходном списке) совпадает с самим элементом, добавление после каждого элемента его копии, уменьшенной в 10 раз (целой части полученного частного).
- 18 Нахождение наименьшего числа, кратного десяти; удаление всех двузначных натуральных чисел, добавление после трехзначных натуральных чисел копии.
- 19 Нахождение наибольшего числа, кратного десяти; удаление всех чисел, в которых разряд десятков и единиц совпадает, добавление после каждого числа, пропорционального последней цифре частного от деления числа на последнюю цифру.
- 20 Нахождение количества чисел, у которых последняя цифра – 5; удаление всех чисел, стоящих после числа, на единицу большего (в исходном списке), добавление после положительных чисел элемента списка, равного единице.
- 21 Нахождение произведения однозначных чисел списка; удаление чисел, меньших предыдущего (в исходном списке), добавление после каждого элемента остатка от деления его на 3.

- 22 Нахождение суммы всех четных чисел списка; удаление чисел, в которых разряд сотен равен разряду единиц, вставление после каждого элемента списка суммы его и первого элемента списка.
- 23 Вставить после каждого элемента списка сумму всех элементов списка; удаление элементов, равных нулю.
- 24 Вставить после четных элементов списка произведение всех элементов списка; удаление элементов, пропорциональных трем.
- 25 Вставить после каждого третьего элемента списка (начиная с первого) максимальное число списка; удаление тех элементов, которые были максимальны в исходном списке.
- 26 Вставить после максимальных элементов списка их копии; удаление четных чисел.
- 27 Добавление после каждого числа списка очередного числа Фибоначчи; после чего удаление максимальных элементов списка.
- 28 Добавление после каждого элемента списка суммы всех элементов исходного списка до текущего числа включительно; после этого удаление минимальных элементов списка.
- 29 Добавление после четных элементов списка номера по порядку этого четного числа; после чего удаление всех чисел, содержащие максимальный разряд единиц.
- 30 Добавление после элементов, кратных трем, частного от деления этих элементов на три; после чего удаление элементов списка, на которые делится максимальный элемент.
- 31 Добавление после элементов, в которых совпадают последний и предпоследний разряд, значения последнего разряда; после чего удаление элементов списка, которые делятся на минимальный элемент.
- 32 Добавление после трехзначных элементов минимального положительного элемента списка; удаление четных чисел из списка.

8 Практическая работа №8. Автоматизация тестирования GUI (2 часа)

С помощью средства AutoIt проведите тестирование любой формы в вашем приложении (например, из курсовой работы). Форма должна содержать хотя бы три активных элемента управления.

9 Практическая работа №9. Автоматизация тестирования WEB-интерфейса (2 часа)

С помощью средства Selenium проведите тестирование любой страницы на любом же сайте (где есть хотя бы три активных элемента управления).

10 Практическая работа №10. UNIT-тестирование (2 часа)

Разработайте 5 UNIT-тестов для программы, разработанной при выполнении практической работы №2, с помощью check (check.sourceforge.net), попутно добавив в Makefile цель test с целью их прохождения.

11 Практическая работа №11. Статический анализ кода (2 часа)

Скомпилируйте все разработанные программы в рамках практических работ 10, 7, 1 и 2 с использованием транслятора clang и проанализируйте с помощью scan-build. В отчете отразите все найденные уязвимости, процесс их устранения (то есть их надо исправить). В случае, если исправление нерационально, укажите причину этого.

12 Практическая работа №12. Средства управления требованиями (2 часа)

Разработайте список требований с использованием специального инструментального средства (например, <http://www.am-programs.ru/ReqDownload.shtml>) либо к вашей курсовой работе, либо к дипломному проекту; либо к любому другому проекту аналогичного объема.

13 Практическая работа №13. Системы отслеживания ошибок (2 часа)

Работа засчитывается при условии учета имеющихся ошибок и запросов на новую функциональность в любом вашем проекте или при выполнении практических работ.

14 Практическая работа №14. Средство создания документации (4 часа)

С использованием технологии DOCBOOK создайте часть пользовательской документации (2-3 страницы) для одной из ваших курсовых работ.

15 Практическая работа №15. Инструментальные средства проектирования (4 часа)

С помощью инструментального средства рисования диаграмм, поддерживающего автоматическое преобразование диаграммы в исходный код, нарисуйте диаграмму классов для следующей задачи, осуществите ее автоматическое преобразование в шаблон исходного кода, дополните исходный код до работающего примера.

Примеры ИС: *umbrello*; *Modelio*

- 1 Товары Интернет-магазина – книги и диски. Все товары определяются ценой, книги имеют название, автора, количество страниц; диски – название, количество треков. Выведите на экран все товары со стоимостью меньше 100 рублей.
- 2 Преподаватели определяются ФИО. Для тех, кто имеет диссертацию дополнительно вводится ее название; для остальных – стаж работы. Вывести всех преподавателей, у которых ФИО начинается на букву «А».
- 3 Телефоны определяются названием модели. Проводные телефоны дополнительно определяются типом номеронабирателя (диск или кнопки); а беспроводные – дальностью действия радиосигнала. Вывести все телефоны, название которых начинается на «А».
- 4 Покатушки определяются названием и расстоянием. Однодневные катушки дополнительно определяются плановым временем поездки (в часах). Многодневные катушки определяются количеством дней и категорией сложности похода (от 1 до 6). Вывести все покатушки длиной более 100 км.
- 5 Музыкальная композиция определяется названием и композитором. Дополнительно для песни указывается автор стихов. Выведите информацию о всех композициях, у которых композитор начинается на букву «А».
- 6 Олимпиада определяется названием. Если олимпиада участвует в программе приема в ВУЗы дополнительно указывается уровень олимпиады (1–3), если олимпиада – этап всероссийской, то указывается название этапа (школьная, окружная, региональная, всероссийская), в остальных случаях – размер призового фонда. Выведите все олимпиады, название которых начинается на букву «А».
- 7 Проездной билет определяется стоимостью. Билет на количество поездок определяется количеством поездок. Билет на неограниченное количество поездок определяется сроком действия (1 день, 5 дней, 10 дней, 15 дней, месяц, три месяца, 6 месяцев, год). Выведите информацию о билетах, стоимостью меньше 300 рублей.
- 8 Информация о студенте определяется ФИО. Для студентов, не имеющих автомата, указывается балл, полученный на экзамене (2–5); для студентов, имеющих автомат указывается основание (олимпиада или контрольные работы). В случае, если контрольная работа – то также указывается средний балл за к/р. Выведите всю информацию о студентах с фамилией, начинающейся на буквы от А до К.
- 9 Сотовый телефон определяется названием. Для смартфонов указывается операционная система. А для других телефонов – наличие браузера. Выведите информацию о телефонах, название которых содержит слово «Nokia».
- 10 Куртка определяется названием модели, наличием капюшона. Для мембранных курток указывается степень водонепроницаемости (число в мм рт. ст.), для остальных – наличием пропитки. Выведите информацию обо всех куртках, имеющих капюшон.
- 11 Жесткий диск определяется названием и емкостью. Внешние жесткие диски определяются дополнительно наличием системы, смягчающей последствия падения. Внутренние жесткие диски – размером (2.5/3.5 дюйма). Выведите информацию о дисках, емкостью больше 200 Гб.
- 12 Велосипед определяется названием модели. Горному велосипеду соответствует количество скоростей, BMX – тип конструкции (фривил, кассетная, фрикостер). Выведите информацию обо всех велосипедах, содержащих в названии «Norco».
- 13 Электронная книга определяется названием и размером экрана. Для EInk-дисплея указывается поколение (pearl, vizplex); для LCD – количество поддерживаемых цветов. Выведите информацию о всех книгах с размером экрана не менее 7 дюймов.
- 14 GPS определяется названием, диагональю экрана. Для переносных GPS указывается наличие велосипедного крепления; для автомобильных – поддержка отображения пробок и наличие радар-детектора. Выведите информацию обо всех GPS с размером экрана менее 7 дюймов.

- 15 Пылесос определяется названием модели. Для обычного пылесоса указывается мощность, для пылесоса-робота – размер убираемого помещения и количество виртуальных стен. Выведите информацию обо всех пылесосах, содержащих в названии слово Indesit.
- 16 Туры определяются названием. Для пляжного тура указывается тип пляжа (галечный, песок); для экскурсионного – количеством экскурсий. Выведите информацию обо всех турах, содержащих слово Египет.
- 17 Язык программирования определяется названием. Алгоритмические языки определяются поддержкой ООП (отсутствует, на классах, прототипная), остальные языки – типом (функциональный, логический, стиль). Выведите информацию обо всех языках, название которых начинается с буквы «А».
- 18 Контрагенты определяются названием. Индивидуальные предприниматели дополнительно определяются наличием счета в банка, а юридические лица – формой организации (ООО, ОАО, ЗАО). Выведите информацию обо всех контрагентах, название которых начинается с буквы «А».
- 19 Счет в банке определяется номером. Для текущего счета указывается плата за обслуживание, для сберегательного счета – проценты годовых и наличие капитализации. Выведите информацию обо всех счетах, номер которого начинается с 408178...
- 20 Автомобильная дорога определяется названием и километражом. Бесплатная дорога определяется статусом автомагистрали (автомагистраль или нет), а платная – стоимостью за километр для обычных пользователей. Выведите информацию о дорогах, длина которых менее 100 км.
- 21 Офисное здание определяется адресом. В случае наличия стоянки указывается количество машиномест и стоимость аренды за месяц. Выведите информацию о зданиях, в адресе которых присутствует слово Тверская.
- 22 Товары Интернет-магазина – GPS-навигаторы и карты. Все товары определяются ценой и названием, GPS-навигаторы имеют назначение (ручной, автомобильный) и признак возможности загрузки карт; карты – размером (в Мб). Выведите информацию о всех товарах со стоимостью менее 4000 рублей.
- 23 Товары Интернет-магазина – чай и кофе. Все товары определяются ценой, названием и весом, кофе – типом (растворимый, молотый, в зернах), чай – типом (черный, зеленый). Выведите информацию о всех товарах с весом менее 150 г.
- 24 Объекты продаваемые в коттеджном поселке: участки (определяются площадью, стоимостью, наличием подряда), дома (определяются этажностью, площадью и стоимостью). Выведите все объекты со стоимостью меньше 1000000 рублей.
- 25 Вопросам теста соответствует формулировка и количество баллов за правильный ответ. Вопросам с вариантами правильных ответов соответствует 4 варианта ответа и номер правильного ответа; остальным вопросам – формулировка правильного ответа. Выведите все вопросы, оцениваемые в 10 баллов и выше.
- 26 Слова определяются собственно словом. Для существительных указывается род, для глаголов – спряжение. Выведите информацию обо всех словах, начинающихся на букву «А».
- 27 Операционная система определяется названием. Для операционной системы на базе Linux указывается название менеджера пакетов; для остальных – стоимость лицензии. Вывести все операционные системы, у которых название начинается на букву «А».
- 28 Рюкзаки определяются названием модели и емкостью. Для городских рюкзаков указывается наличие «вентилируемой спины» для походных – количество отделений и наличие крепления для трекинговых палок. Вывести информацию обо всех рюкзаках, в названии которых присутствует слово «Trek».
- 29 Автостоянка определяется названием, количеством машиномест. Для крытой автостоянки указывается количество этажей. Для открытой стоянки – наличие охраны. Вывести информацию обо всех автостоянках с количеством мест больше 20.
- 30 Партия определяется названием. Для тех партий, что финансируются из бюджета указывается размер ассигнований, а для остальных – количество депутатов в каких-либо представительных органах власти. Вывести информацию обо всех партиях с названиями, начинающимися на буквы от «А» до «К».

16 Практическая работа №16. Средства обратной разработки (2 часа)

Путем декомпиляции программы, выданной преподавателем, определите по какой формуле осуществляется расчет программы.

Декомпиляцию можно осуществить с помощью утилиты objdump (опция -d).

17 Практическая работа №17. Средства автоматизированного проектирования (4 часа)

Создайте программу на языке Си, проверяющую корректность строки (на языке Паскаль) с использованием программ lex (по желанию и yacc).

- 1 Программа проверки правильности оператора case с переменной-селектором типа char (без else), в который вложены операторы присваивания переменной строковой константы.
- 2 Программа проверки правильности оператора присваивания, в правой части которого допустимы операции сложения, вычитания, умножения, деления, переменные, целые и вещественные числа (включая показательную форму).
- 3 Программа проверки правильности описания массива, у которого индексы могут иметь тип-название (например, boolean), ограниченный тип (для integer), а тип-элемента – название типа (идентификатор).
- 4 Программа проверки правильности описания массива, у которого индексы могут иметь тип-название (например, boolean), перечислимый тип, а тип-элемента – ограниченный тип для integer.
- 5 Программа проверки правильности описания массива, у которого индексы могут иметь тип-название (например, boolean), ограниченный тип (для char), а тип-элемента – перечислимый тип.
- 6 Программа проверки описания перечислимого и ограниченного типа (для integer, char). Строка должна начинаться со слова type.
- 7 Программа проверки правильности оператора write(ln), у которого в качестве аргументов могут участвовать строковый литералы, целые и вещественные числа, выражения над числами, соединенные операциями сложения, вычитания, умножения и деления. Может также быть указан формат вывода.
- 8 Проверка правильности последовательности операторов read(ln), присваивания и write(ln). У writeln возможные аргументы – строковые литералы; у read(ln) – названия переменных, элементы массивов. У оператора присваивания слева название переменной или элемент массива, справа число (целое или вещественное).
- 9 Проверка правильности оператора for, у которого начальным и конечным значениями могут быть как целые числа, так и символы, а тело цикла – оператор write(ln), у которого аргументы – целые и вещественные числа и переменные.
- 10 Проверка правильности оператора if, у которого условие имеет вид: <переменная><знак><число>, при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; число – целое или вещественное. Как в части then, так и в else (может быть опущен) указывается оператор write(ln) с аргументами – переменными и строками.
- 11 Проверка правильности последовательности операторов присваивания, правая часть которых – выражения, в которых используются литералы типа «множество» со значениями типа integer, переменные и операции +, -, *.
- 12 Проверка правильности последовательности операторов присваивания, правая часть которых – выражения, в которых используются литералы типа «множество» со значениями типа char, переменные и операции +, -, *.
- 13 Проверка правильности оператора while, у которого условие имеет вид: <переменная><знак><число>, при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; число – целое или вещественное. Телом цикла является либо снова оператор while, либо оператор присваивания, в правой части которого могут быть представлены целые числа, либо переменные.
- 14 Проверка правильности оператора if, у которого условие имеет вид: <переменная><знак><строка>, при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно. Как в части then, так и в else (может быть опущен) указывается оператор присваивания, правая часть которого – выражение, содержащее переменные, целые и вещественные числа и знаки операций +, -, *, /.
- 15 Проверка правильности оператора repeat..until, у которого условие имеет вид: <переменная><знак><число>, при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; число – целое или вещественное. Телом цикла является либо снова оператор repeat..until, либо оператор присваивания, в правой части которого могут быть представлены целые числа, либо переменные.
- 16 Программа проверки правильности оператора case с переменной-селектором перечислимого типа (без else), в который вложены операторы присваивания переменной константы типа integer.
- 17 Проверка правильности последовательности операторов присваивания, правая часть которых – выражения, в которых используются литералы типа «множество» со значениями перечислимого типа, переменные и операции +, -, *.
- 18 Проверка правильности оператора for, у которого начальным и конечным значениями могут быть значения перечислимого типа, а тело цикла – оператор присваивания, в правой части которого записана числовая константа.

- 19 Проверка правильности оператора `if`, у которого условие – это логическое выражение над переменными логического типа. Как в части `then`, так и в `else` (может быть опущен) указывается оператор `write(ln)` с аргументами – переменными и числами.
- 20 Проверка правильности оператора `while`, у которого условие – это логическое выражение над переменными логического типа. Телом цикла является оператор присваивания, правая часть которого – выражение, содержащее целые константы и операции сложения, вычитания, умножения и деления.
- 21 Проверка правильности оператора `repeat..until`, у которого условие – это логическое выражение над переменными логического типа. Телом цикла является оператор присваивания, правая часть которого – выражение, содержащее целые константы и операции сложения, вычитания, `mod` и `div`.
- 22 Проверка правильности оператора `repeat..until`, у которого условие имеет вид: `<переменная><знак><строка>`, при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; строка – литерал типа `string`. Телом цикла является либо снова оператор `repeat..until`, либо оператор `writeln`, аргументами которого являются целые и вещественные числа, возможно, с указанием формата.
- 23 Проверка правильности оператора `while`, у которого условие имеет вид: `<переменная><знак><строка>`, при этом знак – это знак больше, меньше, больше или равно, меньше или равно, равно, не равно; строка – литерал типа `string`. Телом цикла является либо снова оператор `while`, либо оператор `writeln`, аргументами которого являются выражения, содержащие целые числа, имена переменных и операции `+`, `-`, `*`, `/`.
- 24 Программа проверки правильности оператора `write(ln)`, у которого в качестве аргументов могут участвовать строковый литералы и переменные, соединенные операцией сложения; целые числа и переменные, соединенные операциями сложения, вычитания, `mod` и `div`. Может также быть указан формат вывода.
- 25 Проверка правильности последовательности операторов присваивания, правая часть которых – выражения, в которых используются строковые литералы и переменные, соединенные знаком `+`; а также выражения, в которых используются литералы типа `integer`, переменные и операции сложения, вычитания, `div` и `mod`.