

1.Строковый тип

1.Дана последовательность, содержащая от 2 до 30 слов, в каждом из которых от 2 до 10 латинских букв; между соседними словами - не менее одного пробела, за последним словом точка. Напечатать все слова, отличные от последнего слова, предварительно преобразовав каждое из них по следующему правилу:

- перенести первую букву в конец слова;

2. Условие задачи 1:

-удалить из слова первую букву;

3.Условие задачи 1:

-удалить из слова последнюю букву;

4.Условие задачи 1:

-удалить из слова все последующие вхождения первой буквы;

5. Условие задачи 1:

- удалить из слова все предыдущие вхождения последней буквы;

6. Условие задачи 1:

-оставить в слове только первые вхождения каждой буквы;

7. Условие задачи 1:

-если слово нечетной длины, то удалить его среднюю букву.

8.Дана последовательность, содержащая от 2 до 50 слов, в каждом из которых от 1 до 8 строчных латинских букв; между соседними словами - не менее одного пробела, за последним словом - точка. Напечатать те слова последовательности, которые отличны от последнего слова и удовлетворяют следующему свойству:

- слово симметрично;

9. Условие задачи 8:

-буквы слова упорядочены по алфавиту;

10. Условие задачи 8:

-слово совпадает с начальным отрезком латинского алфавита (a, ab, abc, т.д.); учесть, что в диапазоне 'a'..'z' могут быть литеры, отличные от латинских букв;

11.Условие задачи 8:

-слово совпадает с конечным отрезком латинского алфавита (z, yz, xuz и т.д.);

12. Условие задачи 8:

-в слове нет повторяющихся букв;

13. Условие задачи 8:

-каждая буква входит в слово не менее 2-х раз;

14. Условие задачи 8:

-в слове гласные буквы (a,e,i,o,u) чередуются с согласными.

15. Даны натуральное число n , символы S_1, \dots, S_n . Будем рассматривать слова, образованные символами, входящими в последовательность S_1, \dots, S_n группы символов, разделенные одним или несколькими пробелами, назовем словами.

- Удалить все слова с нечетными порядковыми номерами и перевернуть слова с четными номерами.

16. Условие задачи 1:

- Удалить из S_1, \dots, S_n все слова, в которых встречается не более 2-х различных букв.

17. Условие задачи 1:

- Удалить из S_1, \dots, S_n все слова, оканчивающиеся группой букв, "кая" или "кое".

18. Даны натуральное число n , символы S_1, \dots, S_n . Заменить в последовательности каждую группу "child" словом "children".

19. Даны натуральное число n , символы S_1, \dots, S_n . Исключить из последовательности группы символов, расположенные между скобками (.). Сами скобки тоже должны быть исключены.

Предполагается, что внутри каждой пары скобок нет других скобок.

20. Даны натуральное число n , символы S_1, \dots, S_n . Группы символов, разделенные пробелами, будем называть словами. Преобразовать данную последовательность, заменяя всякое вхождение слова "это" на слово "то".

21. Даны символы S_1, \dots, S_n . Известно, что символ S_1 отличен от пробела и что среди S_1, S_2, \dots имеется хотя один пробел. Рассматриваются S_1, \dots, S_n - символы, предшествующие первому пробелу (n заранее неизвестно). Преобразовать последовательность.

- Удалить из нее все символы, не являющиеся буквами (латинскими).

22. Условие задачи 21:

- Заменить все малые буквы одноименными большими.

23. Условие задачи 21:

- Удалить все символьные являющиеся буквами или цифрами, и заменить каждую большую букву одноименной малой.

24. Условие задачи 21:

- Удалить из каждой группы идущих подряд цифр, в которой более 2-х цифр и которой предшествует точка, все цифры, начиная с третьей (например, $ab + 0.1973 - 1.1$ в $ab + 0.19 - 1.1$).

25. Условие задачи 21:

-удалить из каждой группы цифр, которой не предшествует точка, все начальные нули (кроме последнего, если за ним идет точка).

26. Дан текст из 60 литер. Напечатать этот текст, подчеркивая (ставя минусы в соответствующих позициях следующей строки) все входящие в него заглавные и строчные русские буквы.

27. Дана строка литер ($n < 255$). Заменить все вхождения "ph" на "f", а все вхождения "ed" на "inq".

2. Типизированные файлы

1. Дан файл SCHOOL, в котором содержится информация о школьниках:

1) Ф.И.О.

2) год рождения, месяц, число

3) оценки экзаменов

а) отсортировать записи по возрастанию среднего экзаменационного балла;

б) по запросу по Ф.И.О. - выдать средний балл;

в) распечатать ведомость школьников со средним баллом меньше 3.

2. Во входном файле DOLG записаны абоненты ГТС:

1) фамилия

2) адрес

3) номер телефона

4) долг по оплате

а) отсортировать записи по возрастанию долга ГТС;

б) по фамилии _ адресу выдать долг;

в) распечатать ведомость должников.

3. Во входном файле содержится информация об итогах зимней сессии на 1 курсе:

1) фамилия

2) номер группы

3) оценка 1

оценка 2

оценка 3

оценка 4

а) вывести записи со средним баллом больше 4, а также по запросу - список группы с оценками и средним баллом;

- б) вывести записи тех студентов, которые имеют больше двух долгов, а на запрос по фамилии и номеру группы - оценки студента;
- в) отсортировать сведения о студентах, не имеющих отметок ниже 4 в алфавитном порядке;
- г) распечатать записи о тех студентах, средний балл которых больше 4.5, а отдельным списком - неуспевающих (хотя бы одна неудовлетворительная оценка).

4. Дан файл КАТ, содержащий сведения о книгах:

- 1) фамилия автора
- 2) название
- 3) год издания
- 4) цена книги
- а) распечатать записи с заданным годом издания, упорядоченные по алфавиту, а если таковых нет, то выдать соответствующее сообщение;
- б) отсортировать записи в порядке возрастания цены книги.

5. Дан файл f, содержащий сведения о веществах:

- 1) название
- 2) удельный вес
- 3) проводимость (проводник, полупроводник, изолятор)
- а) выбрать данные о проводниках с заданным удельным весом;
- б) выбрать данные об изоляторах, удельный вес которых находится в заданном интервале.

6. Дан файл Р абонентов коммутатора за текущий год. Сведения:

- 1) абонентский номер
- 2) фамилия
- 3) адрес
- 4) сумма долга
- 5) дата последнего платежа
- а) вывести записи должников, у которых дата платежа меньше заданной;
- б) отсортировать записи в порядке возрастания даты последнего платежа.

7. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан файл *багаж*, содержащий сведения о багаже нескольких пассажиров. представляют собой запись с двумя полями: одно поле целого типа (количество вещей) и одно – действительное (вес в килограммах).

- а) Найти багаж, средний вес одной вещи в котором отличается не более, чем на 0.3кг от общего среднего веса одной вещи.
- б) Найти число пассажиров, имеющих более двух вещей и число пассажиров, количество вещей которых превосходит среднее число вещей.

- В) Выяснить, имеется ли пассажир, багаж которого состоит из одной вещи весом менее 30кг.
8. Упорядочить сведения о багаже, записанные в файле *багаж* (см. предыдущую задачу) по возрастанию веса багажа. Предполагается, что число пассажиров, зарегистрировавших багаж, известно заранее и равно n (некоторая константа), при этом n не слишком велико.
Указание: Перенести сведения о багаже из файла *багаж* в массив B_1, \dots, B_n , затем упорядочить этот массив, используя то, что для переменных x, y одного и того же комбинированного типа можно использовать оператор присваивания $x:=y$. После этого переписать элементы массива B_1, \dots, B_n в файл *багаж*.
9. Требуется удалить из данного файла *багаж* (см. задачу 7.), сведения о багаже, общий вес вещей в котором меньше, чем 10кг. Использовать вспомогательный файл *f*.
- 10 Дан файл *библ*, содержащий сведения о книгах. Сведения о каждой из книг – это фамилия автора, название и год издания.
а) Найти названия книг данного автора, изданных с 1960 года.
б) Определить, имеется ли книга с названием “Информатика”. Если да, то сообщить фамилию автора и год издания. Если таких книг несколько, то сообщить имеющиеся сведения обо всех этих книгах.
- 11 Дан файл *T*, который содержит номера телефонов сотрудников учреждения: указывается фамилия сотрудника, его инициалы и номер телефона. Найти номер телефона сотрудника по его фамилии и инициалам.
- 12 Дан файл, содержащий отдельные даты. Каждая дата – это число, месяц и год. Найти:
а) год с наименьшим номером
б) все весенние даты
в) самую позднюю дату
- 13 Дан файл *товар*, содержащий сведения об экспортируемых товарах: указывается наименование товара, страна, импортирующая товар, и объем поставляемой партии в штуках. Составить список стран, в которые экспортируется данный товар, и общий объем этого экспорта.
- 14 Дан файл *ассортимент*, содержащий сведения об игрушках: указывается название игрушки (кукла, кубики, конструктор и т.д.), ее стоимость в копейках и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет).
Получить следующие сведения:

- а) Название игрушек, цена которых не превышает 4р. и которые подходят детям 5 лет.
- б) Цену самого дорогого конструктора.
- в) Названия наиболее дорогих игрушек (цена, которой отличается от цены самой дорогой игрушки не более, чем на 1р.).
- г) Названия игрушек, которые подходят как детям 4-х лет, так и детям 10-и лет.
- д) Можно ли подобрать игрушку, любую, кроме мяча, подходящую ребенку 3-х лет, и дополнить мяч так, чтобы суммарная стоимость игрушек не превосходило 5р.

3. Текстовые файлы

1. Составить в диалоговом режиме программу коррективки текстового файла. Команды подаются с клавиатуры.

* D:N - удалить строку с номером N;

* D:N,M - удалить строки с номерами N, N+1,...M;

* I:N - включить после строки N указанную информацию.

<текст>

2. Дан текстовый файл T. Откорректировать его, удаляя пустые строки, пробелы в конце и вначале строк и оставляя между словами по одному пробелу.

3. Дан текстовый файл T, содержащий предложения на русском языке. Выровнять строки по правому краю, вставляя между словами дополнительные пробелы. За образец считать самую длинную строку.

4. Дан текстовый файл F, содержащий программу на языке Паскаль. Проверить эту программу на соответствие числа открывающих и закрывающих круглых скобок. Считать, что каждый оператор занимает не более одной строки файла, но в строке может быть несколько операторов.

5. Имеется текстовый файл V. Написать программу, которая игнорируя исходное деление этого файла на строки, переформатирует его, разбивая на строки так, чтобы каждая строка оканчивалась точкой, либо содержала ровно 60 литер, если среди них нет точки.

6. Дан текстовый файл F. Записать в перевернутом виде строки файла F в файл G. Порядок строк в файлы G должен быть обратным по отношению к порядку строк исходного файла.

7. Текстовый файл G состоит из слов. Откорректировать его строки, удаляя все слова с нечетными номерами и переворачивая с четными.
8. Дан текстовый файл, каждая строка которого является предложением (используется русский алфавит). В каждом предложении указать слово, в котором доля гласных максимальна.
9. Дан текстовый файл F. Длина каждой строки не превосходит 80 символов. Дополнить короткие строки справа символом '.' так, чтобы их длина была 80.
10. Откорректировать данный текстовый файл T, вставляя в конец каждой его строки число N, равное количеству слов в строке. Под N отводится 6 позиций.
11. Дан текстовый файл F. Определить количество пустых строк и напечатать их номера.
12. Дан текстовый файл G. Откорректировать его, удаляя строки, состоящие из одинаковых литер. Напечатать номера удаляемых строк.
13. Дан текстовый файл. Каждая строка – предложение (слова отделены пробелами). Напечатать предложение, в котором количество слов наибольшее.

4. Модульное программирование

- 1 Составить модуль, реализующий алгоритмы матричной алгебры для произвольных числовых матриц (тип REAL):
 - произведение матриц,
 - умножение матрицы на вектор,
 - определить наибольший (наименьший) элемент в L строке матрицы, дополнительно определить местоположение найденного элемента.
 Реализовать алгоритм задания для элементов, удовлетворяющих заданному условию. Условие задается булевой функцией в вызывающей программе.

- 2 Составить библиотеку алгоритмов для построения матриц вида:

$$\begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ & 1 & \dots & 1 & \\ & \textcircled{X} & & & \textcircled{X} \\ & & & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 & 1 \end{pmatrix} \quad \begin{pmatrix} n & & & & \\ n-1 & n & & & \\ & & \textcircled{X} & & \\ & & & \cdot & \\ 1 & 2 & 3 & \dots & n \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 & \dots & n-1 & n \\ 2 & 1 & 2 & \dots & n-2 & n-1 \\ 3 & 2 & 1 & \dots & n-3 & n-2 \\ & \cdot & \cdot & \cdot & & \\ n & n-1 & n-1 & n-2 & \dots & 1 \end{pmatrix}$$

- вычислить сумму и произведение элементов в L строке (столбце), удовлетворяющих заданному условию; Условие задается булевой функцией в вызывающей программе

3 Составить библиотеку алгоритмов для преобразования произвольных числовых матриц (тип REAL):

- определить наибольший (наименьший) элемент матрицы, Реализовать алгоритм задания для элементов, удовлетворяющих заданному условию. Условие задается булевой функцией в вызывающей программе.
- поменять местами столбцы с номерами N1 и N2,
- вычеркнуть L столбец,
- вычеркнуть K строку и L столбец.

4 Составить библиотеку алгоритмов для обработки произвольных числовых матриц (тип REAL):

- умножение матрицы на число.
 - поменять местами строки с номерами N1 и N2,
 - определить наибольший (наименьший) элемент в столбце матрицы.
- Дополнительно определить местоположение найденного элемента. Реализовать алгоритм задания для элементов, удовлетворяющих заданному условию. Условие задается булевой функцией в вызывающей программе.

5 Составить библиотеку алгоритмов для обработки произвольных числовых матриц (тип REAL):

- транспонирование матрицы,
- сложение и вычитание матриц,
- определить строку с наибольшим (наименьшим) произведением элементов, удовлетворяющих заданному условию. Условие задается булевой функцией в вызывающей программе.

6 Составить библиотеку алгоритмов для обработки произвольных числовых матриц (тип REAL):

- определить количество элементов в L строке (столбце), удовлетворяющих заданному условию. Условие задается булевой функцией в вызывающей программе.
- проверить является ли произвольная квадратная числовая матрица :
единичной,
- вычеркнуть L строку из матрицы.

7 Составить модуль, реализующий алгоритмы проверки: является ли произвольная квадратная числовая матрица :

- единичной,

- трехдиагональной,
- магическим квадратом, т.е. такой, в которой суммы элементов во всех строках и столбцах одинаковые,
- симметричной (относительно главной диагонали).
- посчитать количество элементов матрицы, удовлетворяющих заданному условию. Условие задается булевой функцией в вызывающей программе.

8 Составить модуль, реализующий следующие алгоритмы для произвольных квадратных числовых матриц:

- проверить, является ли матрица треугольной, если "да", то определить какой
- привести матрицу к виду верхней треугольной,
- привести матрицу к виду нижней треугольной
- определить наибольший элемент в каждой строке матрицы. Реализовать алгоритм задания для элементов, удовлетворяющих заданному условию. Условие задается булевой функцией в вызывающей программе

9 Составить библиотеку, реализующую следующие алгоритмы для обработки матриц:

- определить скалярное произведение L строки на K столбец (матрица квадратная);
- определить, является ли целая квадратная матрица ортонормированной, т.е. такой, в которой скалярное произведение каждой пары различных строк равно 0, а скалярное произведение каждой строки на себя равно 1;
- вычислить норму матрицы:
 - а) норма матрицы - это наибольшее значение из модулей элементов,
 - б) норма матрицы - это наибольшее значение из сумм модулей элементов строк.
- определить наименьший элемент в столбце. Реализовать алгоритм задания для элементов, удовлетворяющих заданному условию. Условие задается булевой функцией в вызывающей программе

10 Составить модуль, реализующий следующие алгоритмы:

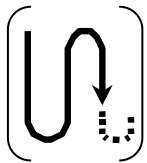
- определить индексы всех седловых точек; (элемент называется седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце или наоборот).
- определить индексы всех локальных минимумов или максимумов (элемент называется локальным минимумом, если он меньше всех своих соседей).

- определить наибольший (наименьший) элемент в строке. Реализовать алгоритм задания для элементов, удовлетворяющих заданному условию. Условие задается булевой функцией в вызывающей программе

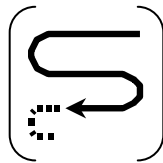
11 Составить модуль, реализующий различные алгоритмы печати матрицы:

- а) по строкам в виде (прямоугольника),
- б) по столбцам (аналогично),

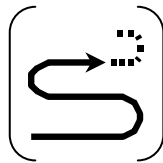
в)



г)



д)



- определить наибольший (наименьший) элемент в столбце. Реализовать алгоритм задания для элементов, удовлетворяющих заданному условию. Условие задается булевой функцией в вызывающей программе

12 Составить модуль для обработки одномерных массивов:

- алгоритмы суммы, произведения, max, min,
- алгоритм определения количества элементов, удовлетворяющих заданному условию Условие задается булевой функцией в вызывающей программе.
- скалярное произведение двух векторов,
- определить, упорядочен ли массив по возрастанию (убыванию), если нет, то упорядочить массив по возрастанию (убыванию).

13 Составить библиотеку алгоритмов для обработки произвольных числовых матриц (тип real):

- Определить норму матрицы

а) $\max_i \sum_j |a_{ij}|$;

б) $\max_j \sum_i |a_{ij}|$;

- - определить максимум среди сумм элементов диагоналей, параллельных побочной диагонали матрицы. В сумму накапливаются те элементы диагоналей, которые удовлетворяют заданному условию. Условие задается булевой функцией в вызывающей программе

14 Составить библиотеку алгоритмов для обработки произвольных числовых матриц (тип real):

- вычислить след матрицы (след матрицы - это сумма диагональных элементов).
- определить столбец с наибольшим (наименьшим) произведением элементов. Реализовать алгоритм задания для элементов, удовлетворяющих заданному условию. Условие задается булевой функцией в вызывающей программе

15 Для произвольных числовых матриц составить библиотеку алгоритмов:

- переставляя строки и столбцы матрицы добиться, чтобы max (min) элемент оказался:
 - а) в верхнем левом углу матрицы;
 - б) в нижнем правом углу матрицы;
- вычислить след матрицы (след - сумма диагональных элементов).
- вычислить сумму элементов матрицы, удовлетворяющих заданному условию. Условие задается булевой функцией в вызывающей программе.

16 Составить модуль для работы с натуральными числами:

- определить, является ли число простым,
- получить делители числа,
- определить, является ли число совершенным, т.е. равным сумме всех своих делителей за исключением самого себя,
- определить, является ли заданное число числом Мерсена (простое число называется числом Мерсена, если оно может быть представлено в виде $2^p - 1$, где p - простое число).

17 Составить модуль для обработки натуральных чисел:

- определить, являются ли два числа дружественными, (два числа называются дружественными, если каждое из них равно сумме все делителей другого, кроме самого этого числа)
- определить, является ли число палиндромом,
- определить, является ли данное число числом Армстронга (натуральное число из n цифр называется числом Армстронга, если сумма его цифр, возведенных в n -ую степень, равна самому числу)

18 Составить модуль для перевода чисел (целых, дробей, смешанных) из десятичной системы счисления в любую p -ичную систему и обратно.

19 Составить модуль для обработки одномерных массивов:

- алгоритмы суммы, произведения, max, min, определения количества элементов, удовлетворяющих заданному условию),
- скалярное произведение двух векторов,

- определить, упорядочен ли массив по возрастанию (убыванию), упорядочить массив по возрастанию (убыванию).

20 Составить библиотеку алгоритмов для обработки и преобразования шкалы S:

- сдвинуть элементы шкалы на K позиций влево
- сдвинуть элементы шкалы на K позиций вправо
- поменять местами соседние четные и нечетные элементы
- поменять местами n и m элементы.

21 Составить библиотеку алгоритмов по обработке текстовых файлов:

- закодировать (скрыть смысл, наглядность текста) текст
- раскодировать текст
- сжатие текста (удаление пустых строк)
- форматирование текста, учитывая левую, правую границы.

22 Составить библиотеку алгоритмов для корректировки текстового файла:

- раздвинуть равномерно строку на K позиций вправо, не меняя левой границы
- равномерно сжать строку на K позиций не меняя левой границы
- удалить строку по её номеру.

23 Составить библиотеку алгоритмов для обработки произвольных числовых матриц (тип real):

- определить строку с наибольшим (наименьшим) произведением положительных элементов
- определить столбец с наибольшим (наименьшим) произведением положительных элементов
- определить наибольший (наименьший) элемент в строке
- определить наибольший (наименьший) элемент в столбце.

24 Составить библиотеку алгоритмов по обработке текстовых файлов:

- напечатать заданное предложение таким образом, чтобы каждое его слово целиком находилось в одной и той же строке распечатки (т.е. избавиться от переносов)
- копирование файла.

25 Составить библиотеку алгоритмов для обработки произвольных числовых матриц (тип real):

- Определить норму матрицы

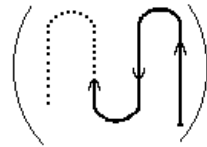
а) $\max_i \sum_j |a_{ij}|$;

б) $\max_j \sum_i |a_{ij}|$;

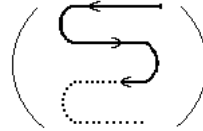
- определить максимум среди сумм элементов диагоналей, параллельных побочной диагонали матрицы.

26 Для произвольных числовых матриц составить библиотеку алгоритмов:
- напечатать элементы заданной матрицы в следующем порядке:

а)



б)



в) найти max (min) среди сумм элементов диагоналей, параллельных главной диагонали.

27 Для произвольных числовых матриц составить библиотеку алгоритмов:
- переставляя строки и столбцы матрицы добиться, чтобы max (min) элемент оказался:

а) в верхнем левом углу матрицы;

б) в нижнем правом углу матрицы;

в) вычислить след. матрицы (след. - сумма диагональных элементов).

5. Динамические структуры данных

- 1 ("Считалка") n ребят располагаются по кругу. Начав отсчет от первого, удаляют каждого k-го, смыкая круг после каждого удаления. Определить порядок удаления ребят из круга.
- 2 Определить, симметричен ли заданный во входном файле текст (за ним следует точка).
- 3 Дана последовательность из не менее чем двух натуральных различных чисел, за которой следует 0. Напечатать в обратном порядке все числа между наибольшим и наименьшим числами этой последовательности.
- 4 Дана непустая последовательность непустых слов из букв; между соседними словами - запятая, за последним словом - точка. Напечатать все слова максимальной длины.

- 5 Дана непустая последовательность слов, в каждом из которых от 1 до 12 строчных латинских букв; между словами - пробел, за последним словом - точка. Напечатать эти слова по алфавиту, указав для каждого из них число его вхождений в эту последовательность.
- 6 Решить предыдущую задачу в предположении, что максимальная длина слов заранее не известна.
- 7 Дана непустая последовательность слов, в каждом из которых от 1 до 8 строчных латинских букв: между словами - пробел, за последним словом - точка. Напечатать эти слова в следующем порядке: сначала - по алфавиту все слова из одной буквы, затем по алфавиту все двухбуквенные слова и т.д.(одинаковые слова печатать по одному разу).
- 8 Решить предыдущую задачу в предположении, что максимальная длина слов заранее не известна.
- 9 Дан текст, оканчивающийся точкой. Среди литер этого текста особую роль играет знак #, появление которого в тексте означает отмену предыдущей литеры текста; k знаков # подряд отменяют k предыдущих литер (если такие есть). Напечатать данный текст, исправленный с учетом такой роли знака # (например, текст XЭ#E##HELLO#LO должен быть напечатан в виде HELLO).
- 10 Дано произвольное натуральное число n. Напечатать все цифры десятичной записи числа n!
- 11 Дана запись многочлена (от переменной X) произвольной степени с целыми коэффициентами, причем его одночлены могут быть и не упорядочены по степеням X, а одночлены одной и той же степени могут повторяться. Возможный пример: $-8x^4 - 74x + 8x^4 + 5 - x^3$. Требуется привести подобные члены в этом многочлене, после чего распечатать его по убыванию степеней X.
- 12 Напечатать содержимое текстового файла t, выписывая литеры каждой ее строки в обратном порядке.
- 13 В текстовом файле записано без ошибок логическое выражение (ЛВ) в следующей форме:
 $\langle \text{ЛВ} \rangle ::= \text{true} | \text{false} | (\neg \langle \text{ЛВ} \rangle) | (\langle \text{ЛВ} \rangle \wedge \langle \text{ЛВ} \rangle) | (\langle \text{ЛВ} \rangle \vee \langle \text{ЛВ} \rangle)$, где знаки \neg, \wedge, \vee обозначают соответственно отрицание, конъюнкцию и дизъюнкцию.
Вычислить(как Boolean) значение этого выражения.

- 14 Использовать очередь и/или стек. В текстовом файле t записан текст, сбалансированный по круглым скобкам:

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

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

Требуется для каждой пары соответствующих открывающей и закрывающей скобок напечатать номера их позиции в тексте, упорядочив пары номеров в порядке возрастания номеров позиций:

- а) закрывающих скобок

Например, для текста $A-(45-F(x))*(B-C)$ надо напечатать а) 8 10; 12 16; 3 17;

- 15 Условие N14

- б) открывающих скобок

б) 3 17; 8 10; 12 16.

- 16 Даны натуральное число n , действительные числа a_1, \dots, a_n заканчивающиеся 0. Преобразовать последовательность a_1, \dots, a_n , расположив в начале отрицательные члены, а затем - неотрицательные. При этом порядок как отрицательных, так и неотрицательных чисел сохраняется прежним.

- 17 Даны натуральное число n , действительные числа a_1, \dots, a_n заканчивающиеся 0. Преобразовать последовательность a_1, \dots, a_n , расположив в начале отрицательные члены, а затем - неотрицательные. При этом порядок отрицательных чисел изменяется на обратный, а порядок неотрицательных сохраняется прежним.

- 18 Даны натуральное число n , действительные числа a_1, \dots, a_n заканчивающиеся 0. Преобразовать последовательность a_1, \dots, a_n , расположив в начале отрицательные члены, а затем - неотрицательные. При этом порядок отрицательных чисел сохраняется прежним, а порядок неотрицательных изменяется на обратный.

- 19 Даны натуральное число n , действительные числа a_1, \dots, a_n заканчивающиеся 0. Преобразовать последовательность a_1, \dots, a_n , расположив в начале отрицательные члены, а затем - неотрицательные. При этом:
порядок тех и других чисел изменяется на обратный.

- 20 Дано натуральное число n , действительные числа a_1, \dots, a_{2n}

Получить:

$\max(\min(a_1, a_{2n}), \min(a_2, a_{2n-1}), \dots, \min(a_n, a_{n+1}))$, n заранее неизвестно.

- 21 Даны натуральное число n , действительные числа a_1, \dots, a_{2n}

Получить:

$\min (a_1 + a_{2n}, a_2 + a_{2n-1}, \dots, a_n + a_{n+1})$, n заранее неизвестно.

- 22 Даны натуральное число n , действительные числа a_1, a_2, \dots, a_{2n} .
Получить последовательность:
 $a_1, \dots, a_{2n}, a_n, \dots, a_{n+1}$
- 23 Даны натуральное число n , действительные числа x_1, \dots, x_n .
Вычислить:
 $(x_1 + x_2 + 2x_n)(x_2 + x_3 + 2x_{n-1}) \dots (x_{n-1} + x_n + 2x_2)$
- 24 Даны натуральное число n , символы s_1, \dots, s_n . Получить последовательность символов, содержащую только последние вхождения каждого символа с сохранением взаимного порядка этих вхождений.
- 25 Даны натуральное число n , символы s_1, \dots, s_m . Известно, что в последовательность s_1, \dots, s_m входит по крайней мере один пробел. Пусть таково, что S_n - это первый по порядку пробел, входящий в s_1, \dots, s_m (n заранее неизвестно). Выполнить преобразования:
 $s_{n+1}, s_{n+2}, \dots, s_m, s_n, \dots, s_1$.
- 26 Даны натуральное число n , целые числа a_1, \dots, a_n . Выяснить, имеются ли среди чисел a_1, \dots, a_n совпадающие.
- 27 Даны натуральное число n , целые числа a_1, \dots, a_{3n} . Выяснить, верно ли, что для всех a_{2n+1}, \dots, a_{3n} имеются равные среди a_1, \dots, a_{2n} .

6. ASCIIZ – строки (строки с нулевым окончанием)

- 1 Перечислите все слова данного предложения, которые состоят из тех же букв, что и первое слово предложения.
- 2 В заданном предложении найти пару слов, из которых одно является обращением другого.
- 3 Для каждого из слов заданного предложения указать, сколько раз оно встречается в предложении.
- 4 Найти самое длинное симметричное слово заданного предложения.
- 5 Расстояние между двумя словами равной длины – это количество позиций, в которых эти слова различаются. В заданном предложении найти пару наиболее далеко удаленных слов

заданной длины.

- 6 Отредактировать заданное предложение, удаляя из него слова-серии, а также те слова, которые уже встречались в предложении раньше.
- 7 Найти множество слов, которые встречаются в каждом из двух заданных предложений.
- 8 Отредактировать заданное предложение, удаляя из него все слова с нечетными номерами и переворачивая слова с четными номерами.
- 9 Заданы два текста. Каждый текст составлен из попарно различных слов. Определить, можно ли получить второй текст из первого удалением некоторых его символов.
- 10 Найти самое длинное общее слово двух заданных предложений.
- 11 Даны два предложения. Найти самое короткое из слов первого предложения, которого нет во втором предложении.
- 12 Среди слов заданного предложения, которые не являются сериями, найти такое, которое имеет наибольшее число вхождений в предложение.
- 13 Проверить, верно ли, что в заданном предложении всякое несимметричное слово имеет четную длину.
- 14 Отредактировать заданное предложение, удаляя из него слова, которые встречаются в предложении заданное число раз.
- 15 Для каждого символа заданного текста указать, сколько раз он встречается в тексте. Сообщение об одном символе должно печататься не более одного раза.
- 16 Заменить окончание ING/ing каждого слова, встречающегося в заданном предложении на ED.
- 17 Отредактировать заданное предложение, удаляя из него все слова, целиком составленные из вхождений не более чем двух букв.
- 18 Задан текст, в котором нет вхождений символов "(" и ")". Выполнить его сжатие, т.е. заменить всякую максимальную подпоследовательность,

составленную из более чем трех вхождений одного и того же символа, на $(K)A$, где A – повторяемый символ, а $K > 3$ – количество его повторений.

- 19 Отредактировать заданное предложение, заменяя всякое вхождение слова вида $ZXY_X \cdot 0$ на $ZY 0$, где $ZX, Y - 0$ подслова, $_3X$ слова $Z X 0$.
- 20 Дан текст. Найти наибольшую длину слов палиндромов. Если слов палиндромов нет, то ответом должно служить число 0.
- 21 Дан текст. Верно ли, что каждое слово, не являющееся палиндромом имеет четную длину.
- 22 Удалить из заданного текста все слова, встречающиеся более двух раз.
- 23 Дан текст. Напечатать все слова, отличные от последнего, предварительно преобразовав каждое из них по следующему правилу:
- первую и последнюю буквы слова поменять местами;
- 24 Дан текст. Напечатать все слова, отличные от последнего, предварительно преобразовав каждое из них по следующему правилу:
- удалить из слова первую и последнюю буквы;
- 25 Дан текст. Напечатать все слова, отличные от последнего, предварительно преобразовав каждое из них по следующему правилу:
- удалить из слова все последующие вхождения первой буквы;
- 26 Дан текст. Напечатать все слова, отличные от последнего, предварительно преобразовав каждое из них по следующему правилу:
- удалить из слова все предыдущие вхождения последней буквы;
- 27 Дан текст. Напечатать все слова, отличные от последнего, предварительно преобразовав каждое из них по следующему правилу:
- оставить в слове только первые вхождения каждой буквы;
- 28 Дан текст. Напечатать все слова, отличные от последнего, предварительно преобразовав каждое из них по следующему правилу:
- если слово нечетной длины, то удалить его среднюю букву.

8. Рекурсивные алгоритмы

1. Построить синтаксический анализатор для понятия $\langle \text{простое выражение} \rangle ::=$

$$\left\{ \begin{array}{l} \langle \text{идентификатор} \rangle \\ (\langle \text{простое выражение} \rangle \langle \text{знак операции} \rangle \langle \text{простое выражение} \rangle) \end{array} \right\}$$
$$\langle \text{идентификатор} \rangle ::= \left\{ \begin{array}{l} \langle \text{буква} \rangle \\ \langle \text{идентификатор} \rangle \left\{ \begin{array}{l} \langle \text{цифра} \rangle \\ \langle \text{буква} \rangle \end{array} \right\} \end{array} \right\}$$
$$\langle \text{знак операции} \rangle ::= + \mid - \mid *$$

2. Построить синтаксический анализатор для понятия “список параметров”

$$\langle \text{Список параметров} \rangle ::= \langle \text{параметр} \rangle \{ \langle \text{параметр} \rangle \}^*$$
$$\langle \text{параметр} \rangle ::= \left\{ \begin{array}{l} \langle \text{имя} \rangle \langle \text{целое число} \rangle \\ \langle \text{имя} \rangle \langle \text{список параметров} \rangle \end{array} \right\}$$
$$\langle \text{имя} \rangle ::= \langle \text{буква} \rangle [\langle \text{буква} \rangle \langle \text{буква} \rangle]$$
$$\langle \text{целое число} \rangle ::= \left\{ \begin{array}{l} \langle \text{целое без знака} \rangle \\ \{ \pm \} \langle \text{целое без знака} \rangle \end{array} \right\}$$
$$\langle \text{целое без знака} \rangle ::= \langle \text{цифра} \rangle \{ \langle \text{цифра} \rangle \}^*$$

3. Построить синтаксический анализатор для понятия “сумма”

$$\langle \text{сумма} \rangle ::= \left\{ \begin{array}{l} \langle \text{целое число} \rangle \\ \langle \text{идентификатор} \rangle \langle \text{знак операции} \rangle \langle \text{сумма} \rangle \end{array} \right\}$$

$$\langle \text{знак операции} \rangle ::= + \mid -$$

(Определения

целого числа и идентификатора см. выше)

4. Написать программу, которая по заданной сумме (определение понятия см. в 3) печатает значение этой суммы. Считать, что идентификатор – это буква латинского алфавита.

5. Построить синтаксический анализатор для понятия “отношение”

$$\langle \text{отношение} \rangle ::= \langle \text{простое выражение} \rangle \left\{ \begin{array}{l} = \\ > \\ < \\ \leq \\ \geq \\ \langle \rangle \end{array} \right\} \langle \text{простое выражение} \rangle \quad (\text{Определение})$$

простого выражения см. в 1)

6. Построить синтаксический анализатор для понятия вещественное число:

$$\langle \text{вещественное число} \rangle ::= \left\{ \begin{array}{l} \langle \text{целое-число} \rangle . \langle \text{целое-без-знака} \rangle \\ \langle \text{целое-число} \rangle . \langle \text{целое-без-знака} \rangle E \langle \text{целое-число} \rangle \\ \langle \text{целое-число} \rangle E \langle \text{целое-число} \rangle \end{array} \right\}$$

$$\langle \text{целое-без-знака} \rangle ::= \langle \text{цифра} \rangle \{ \langle \text{цифра} \rangle \}^*$$

$$\langle \text{целое число} \rangle ::= \left\{ \begin{array}{l} \langle \text{целое-без-знака} \rangle \\ \left\{ \begin{array}{l} + \\ - \end{array} \right\} \langle \text{целое-без-знака} \rangle \end{array} \right\}$$

7. Построить синтаксический анализатор для понятия простое-логическое:

$$\langle \text{простое логическое} \rangle ::=$$

$$\left\{ \begin{array}{l} TRUE \\ FALSE \\ \langle \text{простой-идентификатор} \rangle \\ NOT \langle \text{простое-логическое} \rangle \\ (\langle \text{простое-логическое} \rangle \langle \text{знак-операция} \rangle \langle \text{простое-логическое} \rangle) \end{array} \right\}$$

$$\langle \text{простой идентификатор} \rangle ::= \langle \text{буква} \rangle$$

$$\langle \text{знак-операция} \rangle ::= \left\{ \begin{array}{l} AND \\ OR \end{array} \right\}$$

8. Построить синтаксический анализатор для понятия константное выражение:

$$\langle \text{константное выражение} \rangle ::= \left\{ \begin{array}{l} \langle \text{цифра} \rangle \{ \langle \text{цифра} \rangle \}^* \\ (\langle \text{константное-выражение} \rangle \left\{ \begin{array}{l} - \\ + \\ * \end{array} \right\} \langle \text{знак-операции} \rangle) \end{array} \right\}$$

9. Построить синтаксический анализатор для понятия скобки:

$$\begin{aligned} \langle \text{скобки} \rangle &::= \left\{ \begin{array}{l} \langle \text{квадр} \rangle \\ \langle \text{кругл} \rangle \end{array} \right\} & \langle \text{квадр} \rangle &::= \left\{ \begin{array}{l} [\langle \text{кругл} \rangle \langle \text{кругл} \rangle] \\ + \end{array} \right\} \\ \langle \text{кругл} \rangle &::= \left\{ \begin{array}{l} (\langle \text{квадр} \rangle \langle \text{квадр} \rangle) \\ - \end{array} \right\} \end{aligned}$$

10. Построить синтаксический анализатор для понятия список-списков:

$$\begin{aligned} \langle \text{список – списков} \rangle &::= \langle \text{список} \rangle \{ \langle \text{список} \rangle \}^* & \langle \text{список} \rangle &::= \langle \text{элемент} \rangle \{ \langle \text{элемент} \rangle \}^* \\ \langle \text{элемент} \rangle &::= \langle \text{бува} \rangle \end{aligned}$$

11. Построить синтаксический анализатор для понятия скобки:

$$\begin{aligned} \langle \text{скобки} \rangle &::= \left\{ \begin{array}{l} \langle \text{кругл} \rangle \\ \langle \text{квадр} \rangle \end{array} \right\} \\ \langle \text{кругл} \rangle &::= \left\{ \begin{array}{l} A \\ ((\langle \text{кругл} \rangle) [\langle \text{квадр} \rangle]) \end{array} \right\} \\ \langle \text{квадр} \rangle &::= \left\{ \begin{array}{l} B \\ [[\langle \text{квадр} \rangle] (\langle \text{кругл} \rangle)] \end{array} \right\} \end{aligned}$$

12. Построить синтаксический анализатор для понятия сумма:

$$\begin{aligned} \langle \text{сумма} \rangle &::= \langle \text{целое} \rangle \{ \langle \text{знак – операции} \rangle \langle \text{целое} \rangle \}^* \\ \langle \text{целое} \rangle &::= \langle \text{цифра} \rangle \{ \langle \text{цифра} \rangle \}^* \\ \langle \text{знак – операции} \rangle &::= \{ +, - \} \end{aligned}$$

Например: 021+16 и 22-суммы, а + 1 – не сумма.

13. Построить синтаксический анализатор для понятия простое-выражение:

$$\begin{aligned} \langle \text{простое – выражение} \rangle &::= \left\{ \begin{array}{l} \langle \text{терм} \rangle \\ \langle \text{терм} \rangle \left\{ \begin{array}{l} + \\ - \end{array} \right\} \langle \text{простое – выражение} \rangle \end{array} \right\} \\ \langle \text{терм} \rangle &::= \left\{ \begin{array}{l} \langle \text{множитель} \rangle \\ \{ \langle \text{множитель} \rangle \}^* \langle \text{терм} \rangle \end{array} \right\} \\ \langle \text{множитель} \rangle &::= \left\{ \begin{array}{l} \langle \text{целое – без – знака} \rangle \\ \langle \text{идентификатор} \rangle \\ \langle \text{простое выражение} \rangle \end{array} \right\} \\ \langle \text{целое – без – знака} \rangle &::= \left\{ \begin{array}{l} \langle \text{цифра} \rangle \\ \langle \text{цифра} \rangle \langle \text{целое – без – знака} \rangle \end{array} \right\} \\ \langle \text{идентификатор} \rangle &::= \langle \text{буква} \rangle \langle \text{символ – побел} \rangle \end{aligned}$$