

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ» (МГТУ ГА)

Кафедра прикладной математики

А.С. Куриленок, А.А. Пичугин

**ИНФОРМАТИКА
И ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ.
ИНФОРМАТИКА**

Учебно-методическое пособие
по проведению практических занятий
и выполнению комплексных домашних заданий

*для студентов I курса
направлений 25.03.02, 25.03.03,
специальности 25.05.03
очной формы обучения*

Москва
ИД Академии Жуковского
2018

УДК 004(07)
ББК 6ф7.3
К93

Рецензент:

Коновалов В.М. – канд. техн. наук, доц.

Куриленок А.С.

К93

Информатика и информационные технологии. Информатика [Текст] : учебно-методическое пособие по проведению практических занятий и выполнению комплексных домашних заданий / А.С. Куриленок, А.А. Пичугин. – М.: ИД Академии Жуковского, 2018. – 32 с.

Данное учебно-методическое пособие издается в соответствии с рабочей программой учебной дисциплины «Информатика и информационные технологии. Информатика» по учебному плану для студентов I курса направлений 25.03.02, 25.03.03 специальности 25.05.03 очной формы обучения.

Рассмотрено и одобрено на заседании кафедры 16.10.2018 г. и методического совета 19.10.2018 г.

УДК 004(07)
ББК 6ф7.3

В авторской редакции

Подписано в печать 03.12.2018 г.

Формат 60x84/16 Печ. л. 2 Усл. печ. л. 1,86

Заказ № 397/1112-УМП05 Тираж 100 экз.

Московский государственный технический университет ГА
125993, Москва, Кронштадтский бульвар, д. 20

Издательский дом Академии имени Н. Е. Жуковского

125167, Москва, 8-го Марта 4-я ул., д. 6А

Тел.: (495) 973-45-68

E-mail: zakaz@itsbook.ru

© Московский государственный технический
университет гражданской авиации, 2018

СОДЕРЖАНИЕ

Введение.....	4
1. Алгоритмизация и программирование. Общие сведения и и основные понятия и определения.....	5
1.1. Схемы алгоритмов (программ). Элементы и их графические обозначения.....	6
1.2. Управляющие (типовые) структуры алгоритмов (программ).....	8
1.3. Алгоритм суммирования элементов одномерного массива.....	13
1.4. Алгоритм произведения элементов одномерного массива.....	14
1.5. Алгоритм вычисления факториала.....	15
1.6. Алгоритм ввода (вывода) элементов двумерного массива (матрицы).....	16
1.7. Алгоритм произведения матриц.....	17
1.8. Алгоритм поиска максимального (минимального) элемента одномерного (двумерного) массива.....	19
1.9. Алгоритм сортировки элементов одномерного (двумерного) массива в порядке возрастания (убывания).....	20
2. Комплексное домашнее задание (КДЗ № 2). Содержание, правила и процесс подготовки и выполнения, отчетность, защита.....	21
Литература.....	23
ПРИЛОЖЕНИЕ 1.....	24
ПРИЛОЖЕНИЕ 2.....	25

Введение

Данное учебно-методическое пособие предназначено для студентов I-го курса очной формы обучения по направлениям подготовки: 25.03.02, 25.03.03, специальности 25.05.03 и содержит ряд материалов, изучаемых на практических занятиях 2-го семестра. Вопросы, рассматриваемые в этих материалах, позволяют и способствуют студентам получить более глубокие и полноценные теоретические знания, конкретные практические навыки и умения, необходимые для освоения и усвоения теоретико-практических материалов раздела «Алгоритмизация и программирование».

Приведены материалы по вопросам организации, подготовки и выполнения комплексного домашнего задания КДЗ № 2: объяснено содержание, правила и процедура подготовки и выполнения КДЗ, формы отчетности и защиты. Представлены варианты заданий для КДЗ.

Целью проведения и выполнения практических занятий и комплексных домашних заданий является:

- закрепление теоретико-практических основ пройденного и изученного материала лекций и практических занятий;
- приобретение необходимых практических навыков и умений, необходимых для выполнения комплексных домашних заданий, а также для выполнения и сдачи индивидуальных домашних заданий и контрольных работ.

1. Алгоритмизация и программирование. Общие сведения и основные понятия и определения

Алгоритмизация и программирование – это два основных этапа в процессе решения любой задачи (проблемы) с помощью использования компьютеров (электронно-вычислительных машин - ЭВМ, вычислительных машин - ВМ), компьютерных технологий.

Любую задачу, которую Вы хотите решить (реализовать) с помощью “вычислителя” (компьютера, ЭВМ, ВМ), Вы должны представить в форме, понятной компьютеру. А так как компьютер (ЭВМ, ВМ) может выполнять операции, четко и однозначно описанные, то и соответствующую задачу необходимо представить в четкой и ясной форме, не допускающей двусмысленностей.

Определения.

Алгоритм – четкое конкретное описание (предписание) последовательности действий, которые необходимо выполнить (совершить) для решения задачи (над исходной информацией с целью получения результата).

Алгоритмизация - процесс построения (проектирования, создания, разработки, объяснения, определения) **алгоритма**, то есть определение последовательности действий, необходимых для решения задачи.

Вторая часть решения задачи – **программирование** – состоит в том, что в соответствии с разработанным алгоритмом разрабатывается (пишется) программная реализация (программа, программные коды), которая выполняется, реализуется непосредственно на “вычислителе” (компьютере, ЭВМ, ВМ). Программная реализация алгоритма разрабатывается и пишется на специальном алгоритмическом языке (на конкретном языке программирования в соответствующей конкретной среде программирования), который понятен “вычислителю”, и, следовательно, программная реализация может быть им выполнена.

Программа – программная реализация *алгоритма*, записанная на конкретном языке программирования в конкретной среде программирования.

Программирование – процесс (процедура) написания (разработки, проектирования) программы (программных кодов) на конкретном языке программирования в конкретной среде программирования.

Алгоритм, как правило, представляется на естественном языке, понятном человеку, который будет писать (разрабатывать) программу, либо в виде графической структуры (схемы) алгоритма, которая состоит из неких стандартных символов и базовых “управляющих” структур. Графическое представление алгоритма это графическое изображение процесса решения конкретной задачи в виде графических условных обозначений (символов) и их связей.

Программа (программные коды) пишется на одном из конкретных языков программирования, который понятен и человеку (иначе он не сможет

написать программу) и “вычислителью” (компьютеру, ЭВМ, ВМ) (иначе он не сможет её выполнить).

Решение любой задачи с помощью “вычислителя” (компьютера, ЭВМ, ВМ), как правило, состоит из следующих этапов:

1. Постановка задачи (смысловое описание, математическое описание, формализация).

2. Разработка алгоритма решения задачи (словесное представление, графическое представление).

3. Разработка программы (программных кодов) (программной реализации алгоритма) (составление программы на конкретном языке программирования).

4. Аprobация программы на “вычислителе” (отладка программы, проверка её работы на контрольных и тестовых примерах (проверочные вычисления), окончательное выполнение – “счет”).


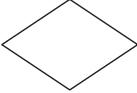
5. Получение окончательного результата решения задачи. Оформление (документирование).


1.1. Схемы алгоритмов (программ). Элементы и их графические обозначения

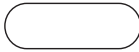
Для наглядного представления алгоритма решения любой задачи используется язык схем (графическое представление), в котором порядок действий, отражающих и раскрывающих суть и содержание алгоритма, указывается и раскрывается с помощью графических элементов и структур.

В таблице 1 приведены основные условные графические обозначения элементов схем алгоритмов и программ.

Таблица 1

Наименование	Обозначение	Функция
Процесс (блок вычислений)		Выполнение операции или группы операций, в результате которых изменяется значение, форма представления данных. Обработка данных любого вида.
Решение (блок проверки условия)		Выбор направления возможного алгоритма или программы в зависимости от некоторых переменных условий. Отображает функцию переключательного типа с одним входом и двумя или более альтернативными выходами, из которых

		только один может быть выбран после вычисления условий, определяемых внутри этого блока.
Модификация		Выполнение операций, меняющих команды или группы команд, изменяющих программу.
Предопределенный процесс		Использование ранее созданных программ или алгоритмов (подпрограмм, модулей).
Соединитель		Межблочный соединитель (используется для переноса процесса, для обрыва линий и их продолжения в другом месте процесса, на других страницах).
Комментарий		Используется для более подробного описания шага, процесса или группы процессов. Описание помещается со стороны квадратной скобки.
Границы цикла (блок – цикл)		Символ, используемый в блоке, состоит из двух частей – начало и конец цикла. Операции, выполняемые внутри цикла, размещаются между символами блока. Условия работы цикла, приращения записываются внутри блока – символа (начала или конца) – в зависимости от типа организации и вида цикла.
Ввод-Вывод		Преобразование данных в форму пригодную для обработки (ввод) и отображения результатов обработки (вывод).

Начало, конец (блок начало-конец) (пуск-останов)		Начало, конец программы, алгоритма. Внутри между ними описываются (отображаются) действия, располагаются структуры.
--	---	--

1.2. Управляющие (типовые) структуры алгоритмов (программ)

Практика программирования показала необходимость научно-обоснованной методологии разработки, создания, проектирования и документирования алгоритмов и программ. Эта методология должна касаться:

- анализа исходной задачи;
- разделения её на достаточно самостоятельные, по возможности простые и понятные части;
- программирования этих частей отдельно, по возможности независимо друг от друга.

Такой методологией, получившей широкое распространение, является *структурное программирование*. По своей сути оно воплощает принципы системного подхода в процессе создания, проектирования, разработки и эксплуатации программного (математико-программного) обеспечения “вычислителя” (компьютера, ЭВМ, ВМ). В основу структурного программирования положены нижеследующие достаточно простые положения:

1. Алгоритм и программа должны составляться поэтапно (по шагам, по инструкциям, по действиям, по операциям).

2. Любая сложная задача должна разбиваться на достаточно простые, легко воспринимаемые и понятные части, каждая из которых имеет только один вход и один выход.

3. Логика алгоритма и программы должна опираться на минимальное число (количество) достаточно простых базовых управляющих структур.

Использование вышеназванных положений позволяет:

- внести определенную систему в труд разработчиков (кодировщиков, программистов) алгоритмов и программ;
- составлять удобочитаемые алгоритмы (и программы), которые легко изучать, контролировать и проверять.

Фундаментом структурного программирования является *теорема о структурировании*. Данная теорема устанавливает следующее: *что как бы сложна ни была исходная задача, схема соответствующего алгоритма (и программы) всегда может быть представлена с использованием весьма ограниченного числа элементарных (типовых) управляющих структур*. Эти элементарные структуры могут соединяться между собой, образуя более

сложные структуры, по тем же самым элементарным схемам. Данные элементарные структуры – базовые управляющие элементарные (типовые) структуры.

Существуют *четыре основные базовые управляющие структуры*, которые обладают функциональной полнотой, то есть любой алгоритм (и программа) может быть реализован(а) в виде композиции этих четырех конструкций. Это линейная, разветвляющиеся, циклическая структуры.

1. Линейная (последовательность).



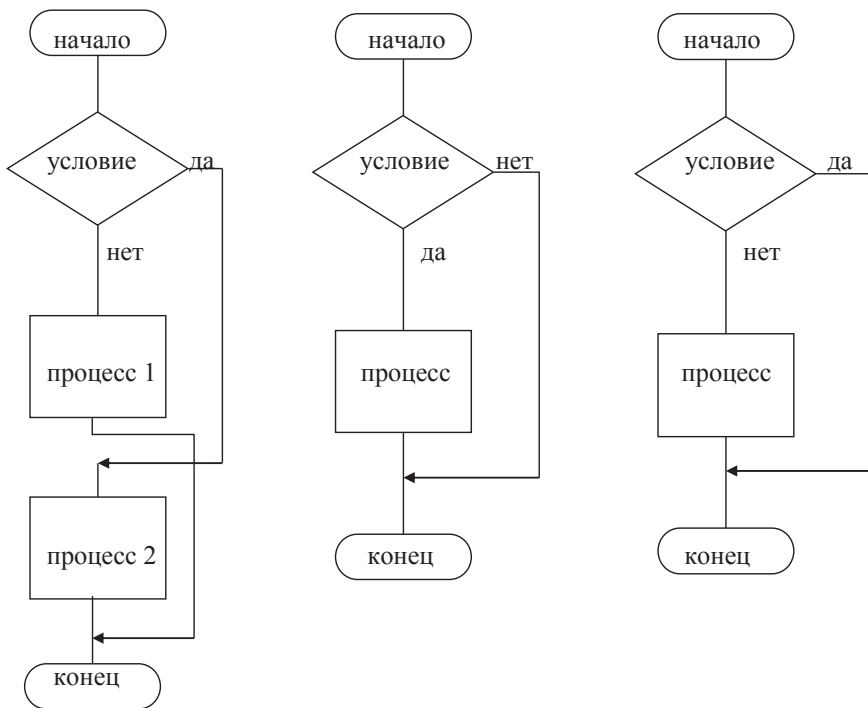
Рис.1.

Алгоритм линейной структуры состоит в последовательном выполнении приведенных в структуре процессов (см. рис. 1).

2. Выбор (ветвление, разветвление).

В алгоритмах разветвляющихся структур весь вычислительный процесс формируется, как правило, в виде двух или трех ветвей (путей) в соответствие с некоторым условием. Данное условие может быть истинным или ложным, тогда работает одна из двух ветвей. Если же условие соответствует (связано) с количественной проверкой выражений отношения “неравенства”: “больше - >”, “меньше - <”, “равно - =”, то ветвление (выбор) будет состоять из трех ветвей. В любом из рассмотренных алгоритмов разветвляющихся структур, состоящих из двух или трех ветвей, реализуется только одна ветвь, связанная с выполнением (не выполнением) условия.

На рис. 2 приведены несколько алгоритмов разветвляющейся структуры.



а) альтернатива

б) коррекция

в) обход

Рис. 2 а, б, в.

3. Вариант (сложное разветвление).

Алгоритмическая структура «вариант» – это разновидность разветвляющихся структур (сложное ветвление). Необходимость подобной структуры возникает, когда нужно сделать выбор более, чем из трех ветвей (путей, вариантов). Фактически – это выбор из «n ветвей» только одного направления (ветви).

На рис. 3 представлена структура «вариант».

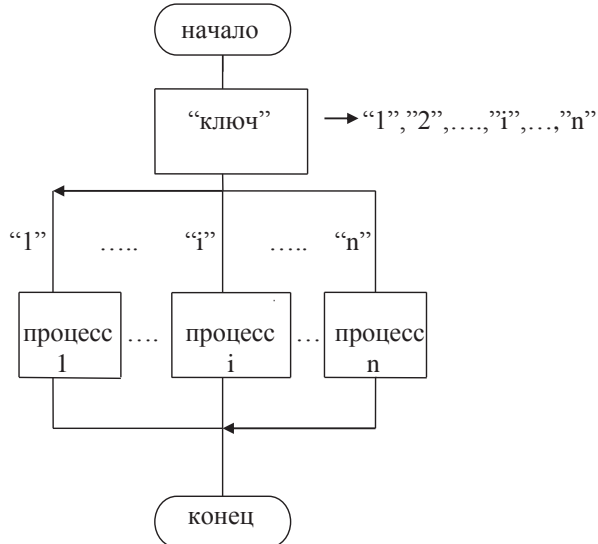


Рис. 3.

4. Цикл (циклическая структура).

Алгоритмы циклической структуры включает в себя регулярно повторяющиеся процессы, называемые «телом цикла». Циклы классифицируются: математические и программные. С точки зрения «повторяемости» циклические процессы подразделяются на:

- арифметические циклы;
- итерационные.

Арифметические циклы – циклы с известным заранее числом повторений. Для этих циклов характерны ряд понятий – параметры цикла, управляющая переменная цикла, начальное и конечное значения управляющей переменной (параметра цикла), значение изменения параметра цикла (модификация параметра, шаг цикла), термин «счетчик цикла». В зависимости от реализации счетчика цикла имеет место еще одна классификация арифметических циклов: 1) циклы с прямым счетчиком; 2) циклы с обратным счетчиком. В этих циклах всегда можно заранее определить число повторений цикла.

Итерационные (итеративные) циклы – циклы с неизвестным заранее числом повторений. В них появляется понятие – итерация, счетчик подсчета шагов (итераций) выполнения цикла. В качестве условия проверки окончания итерационного цикла могут использоваться более сложные (по сравнению с арифметическим циклом) условия окончания цикла – критерии выхода (окончания работы). Итерационные циклы в зависимости от критерия выхода

(окончания работы), расположения условия относительно «тела цикла» подразделяются на: 1) циклы с предусловием; 2) циклы с постусловием.

На рис.4 приведены три варианта циклической структуры.

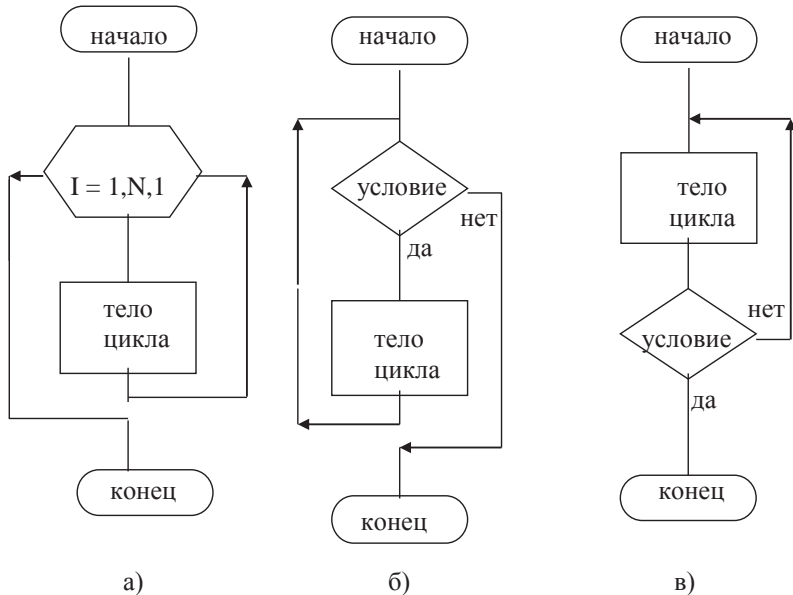


Рис.4 а, б, в.

На рис. 4а представлена структура «арифметический цикл». На рис. 4б представлена структура «итерационный цикл – цикл с предусловием». На рис. 4в представлена структура «итерационный цикл – цикл с постусловием».

Рассмотрим дополнительно ряд информационных материалов, связанных с вопросами проектирования (видами проектирования) алгоритмов (программ).

Если требуется решить задачу с помощью “вычислителя” (компьютера, ЭВМ, VM), то, как ранее говорилось, мы должны разработать **алгоритм** - способ действий, явно и недвусмысленно объясняющий этапы вычислений, которые должны быть выполнены над исходными **данными** задачи, чтобы получить **корректное решение** (когда оно существует) **за конечное время**. Другими словами, алгоритм (как уже указывалось) должен быть записан в виде **программы на языке программирования в соответствующей среде программирования**. До сих пор, мы формулировали алгоритмы сразу на языке схем, вместо того, чтобы тщательно разбивать задачу на более простые части и проверять взаимосвязь между ними, прежде чем, перейти к более детальной спецификации. Такой способ называется **проектированием снизу вверх**.

Если программа создается строка за строкой, то единственный способ ее проверки - выполнить программу на ЭВМ (компьютере) для различных аргументов и посмотреть, будут ли результаты такими, как предполагалось.

Этот путь известен как **тестирование программы**, а процесс изменения программы, с целью исправления ее ошибок - как **отладка**. Если программа выполняется корректно для специально выбранного набора аргументов, то уверенность, с которой можно утверждать, что наша программа, будет выполняться корректно, для **всех** аргументов может быть интуитивно высокой. Но этот метод не может гарантировать корректного выполнения программы во всех случаях. Единственный способ сделать это состоит в тщательном процессе проектирования сверху вниз и включает убедительную демонстрацию того, что **все** вычисления, которая представляет программа, удовлетворяет требовавшимся спецификациям, давая корректные результаты за конечное время.

Практика показала, метод проектирования сверху вниз позволяет разрабатывать качественные программы со значительно меньшим количеством ошибок, чем метод непосредственного (построчного) проектирования.

Суть структурного программирования (как ранее утверждалось) состоит в использовании небольшого набора простых управляющих структур и структур данных, доказательство правильности которых легко осуществить. При этом алгоритм (программа) строится путем вложения команд, действий, инструкций (операторов) одних в другие. При использовании такого подхода ограничивается число связей между частями алгоритма (программы), и алгоритм (программа) становится более понятным. Со структурным программированием связан метод **проектирования программного обеспечения сверху вниз** (нисходящее проектирование).

Сформулируем некоторые положения, связанные с видами проектирования алгоритмов и программ. Нисходящее проектирование (программирование) – реализует процесс «сверху - вниз», использует принцип декомпозиции (“разбиения”), основывается на положении «от общего к частному», используя и реализуя дедуктивный подход (метод дедукции). Восходящее проектирование (программирование) – реализует процесс «снизу-вверх», использует принцип композиции (“собирания”), основывается на положении «от частного к общему», применяя и реализуя индуктивный подход (метод индукции).

1.3. Алгоритм суммирования элементов одномерного массива

Вычисление суммы S одномерного массива размерностью N производится по формуле

$$S = x_1 + x_2 + \dots + x_N = \sum_{i=1}^N x_i .$$

Ввод элементов массива можно описать следующей структурой, приведенной на рис. 5а. Алгоритм суммирования элементов массива

приведен на рис. 5б. Ввод элементов массива показан на этом рисунке обозначением «предопределенный процесс».

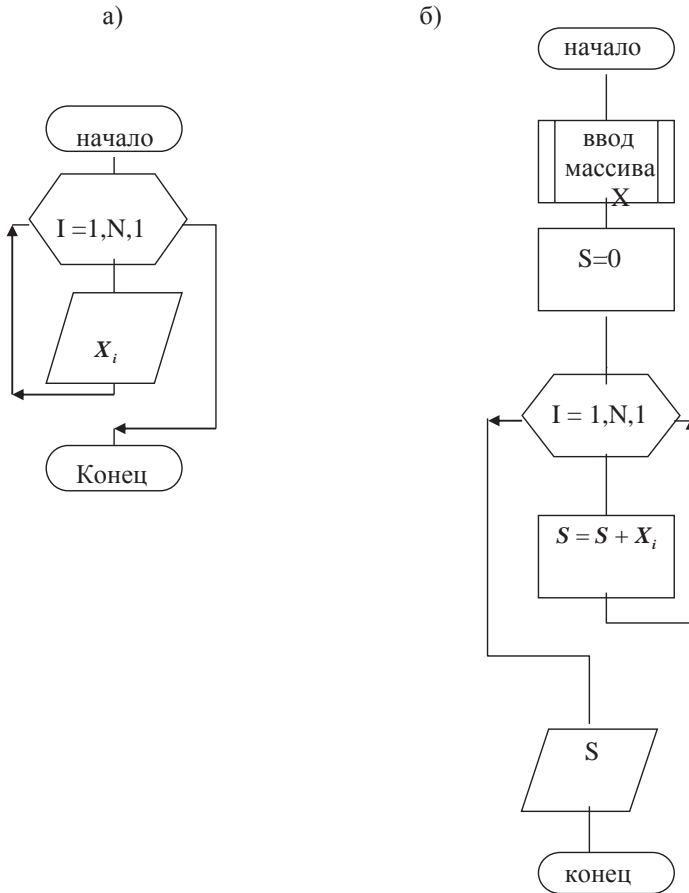


Рис.5 а, б.

1.4. Алгоритм произведения элементов одномерного массива

Вычисление произведения P одномерного массива размерностью N производится по формуле:

$$P = x_1 \cdot x_2 \cdot \dots \cdot x_N = \prod_{i=1}^N x_i .$$

Алгоритм для вычисления произведения элементов массива приведен на рис. 6. Ввод элементов массива показан на этом рисунке обозначением «предопределенный процесс» (см. рис. 5а).

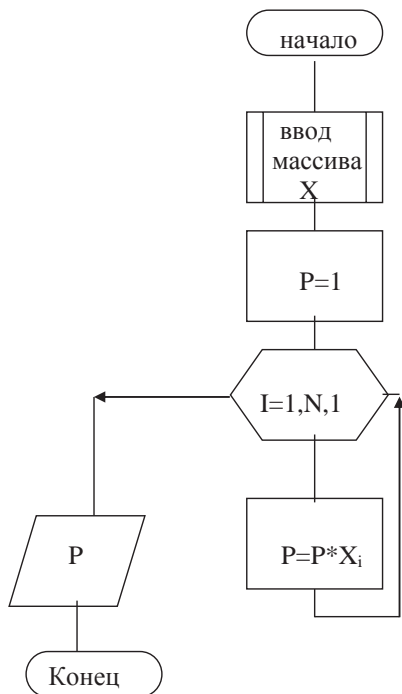


Рис. 6

1.5. Алгоритм вычисления факториала

Вычисление произведения F (факториала) целых чисел от 1 до N производится по формуле

$$F = 1 \cdot 2 \cdot 3 \dots N = N! = \prod_{i=1}^N i.$$

Алгоритм для вычисления факториала приведен на рис. 7.

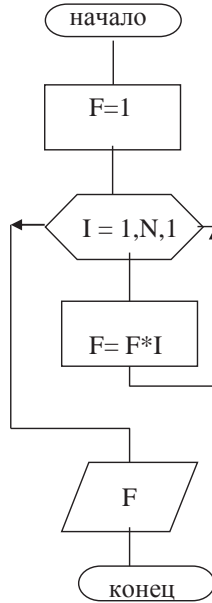


Рис. 7

1.6. Алгоритмы ввода (вывода) элементов двумерного массива (матриц)

Имеем матрицу:

$$A = \begin{pmatrix} a_{11} \dots a_{1j} \dots a_{1m} \\ \dots \dots \dots \\ a_{i1} \dots a_{ij} \dots a_{im} \\ \dots \dots \dots \\ a_{n1} \dots a_{nj} \dots a_{nm} \end{pmatrix} = [a_{ij}]_{N, M} = (N \times M),$$

где N – число строк, M – число столбцов матрицы.

Алгоритм ввода матрицы приведен на рис. 8. Аналогичен вывод матрицы.

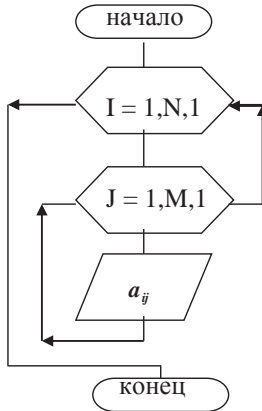


Рис.8

1.7. Алгоритм произведения матриц

Имеем две прямоугольные матрицы:

$$A = (N \times M) \quad \text{и} \quad B = (P \times K),$$

где N – число строк, M – число столбцов матрицы A ,

P – число строк, K – число столбцов матрицы B .

Произведением двух произвольных прямоугольных матриц является прямоугольная матрица:

$$C = A \times B = (N \times M) \times (P \times K) = (N \times K),$$

при условии: число столбцов матрицы A – M должно равняться числу строк матрицы B – P , то есть $M = P$.

$$A = \begin{pmatrix} a_{11} \dots a_{1j} \dots a_{1m} \\ \dots \\ a_{i1} \dots a_{ij} \dots a_{im} \\ \dots \\ a_{n1} \dots a_{nj} \dots a_{nm} \end{pmatrix} = [a_{ij}]_{N,M} = (N \times M),$$

$$B = \begin{pmatrix} b_{11} \dots b_{1j} \dots b_{1k} \\ \dots \\ b_{i1} \dots b_{ij} \dots b_{ik} \\ \dots \\ b_{p1} \dots b_{pj} \dots b_{pk} \end{pmatrix} = [b_{ij}]_{P,K} = (P \times K),$$

$$C = \begin{pmatrix} c_{11} \dots c_{1j} \dots c_{1k} \\ \dots \\ c_{i1} \dots c_{ij} \dots c_{ik} \\ \dots \\ c_{n1} \dots c_{nj} \dots c_{nk} \end{pmatrix} = [c_{ij}]_{N,K} = (N \times K),$$

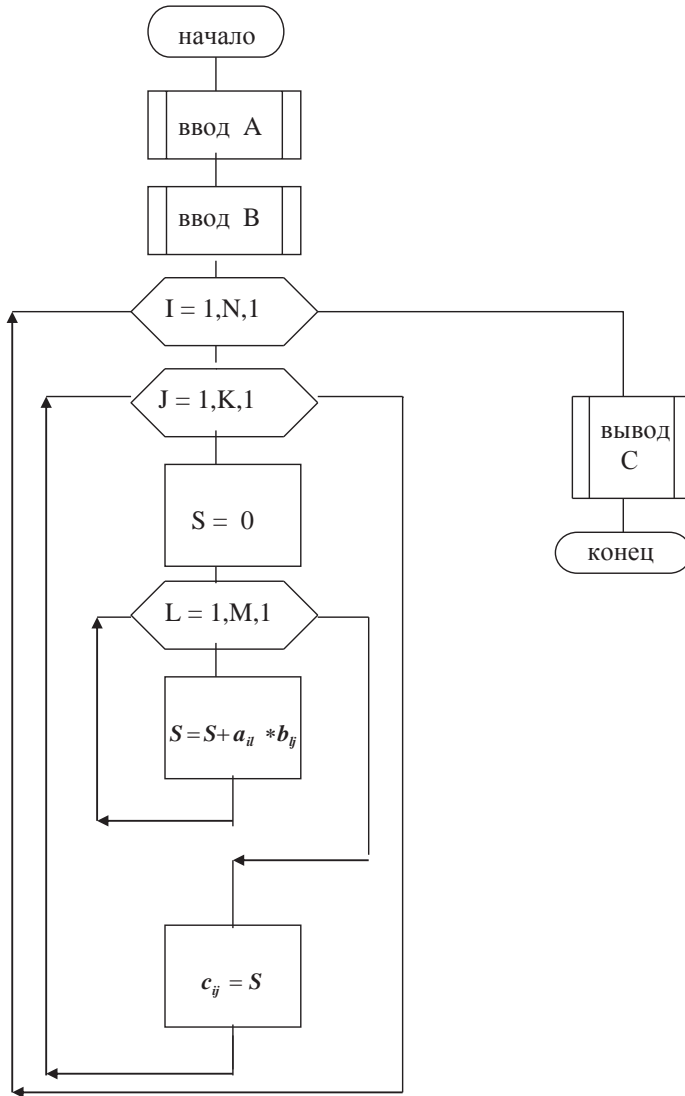


Рис. 9.

На рис. 9 представлен алгоритм перемножения двух прямоугольных матриц (с учетом выполнения условия: $M = P$). Ввод элементов матриц осуществляется с помощью «предопределенного процесса – блока» (см. рис. 8.). Заметим, если две матрицы – квадратные размерностью $(N \times N)$, то в результате вычисления произведения этих матриц получается квадратная матрица той же размерности.

1.8. Алгоритм поиска максимального (минимального) элемента одномерного (двумерного) массива

Дан массив $X = (2,7,3,2,1)$. Очевидно, что максимальный элемент 7, его порядковый номер 2. Тогда процедура поиска максимального элемента может быть следующей. Предположим, что максимальный элемент это первый, обозначим его через X_M . Далее каждый элемент массива, начиная со второго, сравнивается с X_M , и если значение текущего значения больше X_M , тогда X_M принимает новое значение. Алгоритм поиска максимального элемента приведен на рис. 10. Данный алгоритм может быть использован для поиска минимального элемента при замене знака « $<$ » на знак « $>$ » в блоке проверки условия. Используются простые циклические процессы (простые циклы).

Поиск максимального или минимального элементов двумерного массива осуществляется по аналогии. Используются сложные (вложенные) циклы.

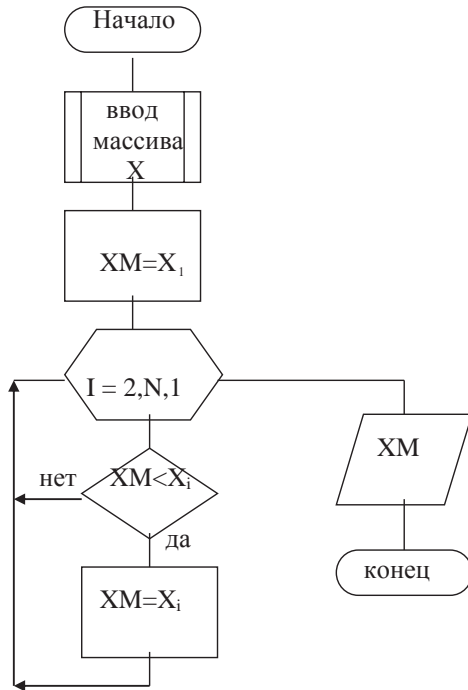


Рис. 10.

1.9. Алгоритм сортировки элементов одномерного (двумерного) массива в порядке возрастания (убывания)

Рассмотрим одномерный массив, приведенный в примере предыдущего пункта 1.8 раздела 1. Требуется провести сортировку элементов этого массива в соответствии с их значениями в порядке возрастания (убывания).

В результате упорядочивания элементов в порядке возрастания этот массив будет такой $X = 1, 2, 2, 3, 7$. Существуют разные алгоритмы сортировки (упорядочивания). Приведем один из них – **метод простого выбора** (метод пузырька). Суть его заключается в следующем:

1. Первый элемент массива сравнивается с остальными.
2. Если на каком-либо этапе сравнения окажется, что $x_1 > x_i$ ($2 \leq i \leq n$), то они меняются местами.
3. В результате на месте x_1 будет стоять наименьший элемент.
4. Теперь сравнивается x_2 с последующими элементами на тех же условиях.
5. После этого просмотра на месте x_2 будет стоять число, большее x_1 , но меньшее всех оставшихся.
6. Далее проводятся аналогичные сравнения для всех остальных элементов до тех пор, пока не будут сравнены два последних.
7. В итоге получим возрастающую последовательность чисел.

Имеем массив $X = (2, 7, 3, 2, 1)$.

После первого прохода имеем 1 7 3 2 2 .

После второго - (1 3 7 2 2) (1 2 7 3 2) .

После третьего - (1 2 3 7 2) (1 2 2 7 3) .

После четвёртого - 1 2 2 3 7 .

Алгоритм сортировки приведен на рис. 11. Данный алгоритм может быть использован для сортировки в порядке убывания при замене знака « $>=$ » на знак « $<=$ » в блоке проверки условия.

Для двумерного массива алгоритм сортировки по убыванию (возрастанию) осуществляется, например, следующим образом: сначала двумерный массив преобразуется в одномерный, затем в нем производится сортировка по убыванию (по возрастанию) аналогично, как осуществлялось, для одномерного массива, после чего, упорядоченный одномерный массив (по убыванию либо по возрастанию) преобразуется обратно в двумерный массив (матрицу).

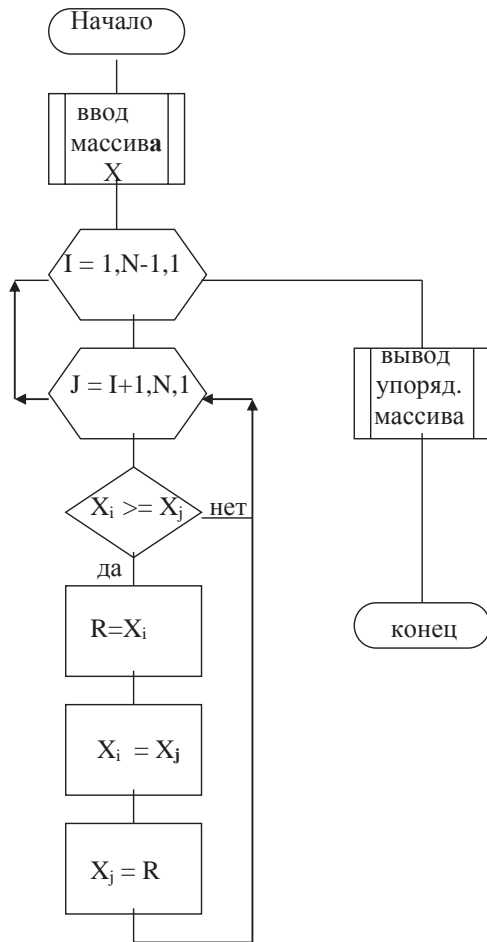


Рис. 11.

2. Комплексное домашнее задание (КДЗ № 2). Содержание, правила и процесс подготовки и выполнения, отчетность, защита

Комплексное домашнее задание № 2 (КДЗ № 2) посвящено вопросам освоения и закрепления теоретико-практических основ (знаний, умений, навыков), связанных с изучением раздела “Алгоритмизация и программирование” для дисциплин: «Информатика и информационные технологии», «Информатика».

В рамках КДЗ №2 студентами решаются конкретные теоретико-практические задачи с использованием компьютерных технологий (в конкретных задачах осуществляется обработка численной и строковой

информации; обработка структур и массивов данных; работа с графикой, в графическом режиме). Студенты разрабатывают, описывают и исследуют алгоритмы решения указанных конкретных задач с последующей апробацией их программных реализаций (программных кодов для событийных процедур) в конкретной среде программирования на конкретном языке программирования (Quick Basic, Free Basic, Visual Basic).

В каждом варианте необходимо выполнить следующее:

1. Разработать алгоритм решения предлагаемой задачи.
2. Представить и нарисовать схему разработанного алгоритма решения.
3. Разработать экранную форму с набором элементов управления.
4. Осуществить программную реализацию (программные коды) разработанного алгоритма в конкретной среде программирования на конкретном языке программирования (типа Quick Basic, Free Basic, Visual Basic) (разработать и написать программу).
5. Получить результаты решения предложенной задачи (апробировать (осуществить выполнение, “счет”) разработанную программу на компьютере).

Пример варианта КДЗ № 2.

ВАРИАНТ 43

Дана таблица сдачи N студентами K экзаменов. Вывести список дисциплин с указанием средних баллов всех студентов по ним (по убыванию средних баллов), а также баллы по предметам и средние баллы лучшего и худшего из сдававших экзамены студентов. Исходная информация вводится с клавиатуры.

В ПРИЛОЖЕНИИ 2 представлен примерный перечень вариантов заданий для КДЗ № 2.

По результатам выполнения КДЗ № 2 студент предоставляет материалы отчета, имеющего нижеследующую структуру и содержание.

Структура отчета:

1. Титульный лист (1 лист).
(В ПРИЛОЖЕНИИ 1 приведён образец).
2. Содержание (оглавление) отчета (1 лист).
3. Задание на КДЗ № 2 (1 лист).
4. Схема алгоритма решения предложенной задачи (1 или несколько листов).
5. Программная реализация (программные коды) разработанного алгоритма решения предложенной задачи (текст программы на конкретном языке программирования – типа Quick Basic, Free Basic, Visual Basic) (1 или несколько листов).
6. Результаты выполнения (результаты “счета”) предложенной задачи на компьютере (“машинные” распечатки) (1 или несколько листов).
7. Материала теории по используемому языку программирования (1 или несколько листов).

Содержание (оглавление) отчета по КДЗ № 2.

1. Задание по КДЗ № 2.
2. Схема алгоритма решения задачи.
3. Программная реализация алгоритма решения.
4. Результаты решения (апробации алгоритма и программы).
5. Материалы теории по языку программирования Free Basic (Quick Basic, Visual Basic).

Литература

1. Информатика. Общий курс: Учебник / А.Н. Гуда и др.; под ред. В.И. Колесникова. – М.: Издательско-торговая корпорация «Дашков и К°»; Ростов н/Д: Наука-Спектр, 2011. – 400 с.
2. Новиков Ф.А. Дискретная математика для программистов: Учебник для вузов.- СПб.: Питер, 2009.- 384 с.
3. Агеев В.Н. Информатика и информационные технологии: Учебно-методическое пособие по проведению практических занятий. – М.: МГТУ ГА, 2017. – 32 с.

ПРИЛОЖЕНИЕ 1
Титульный лист
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ГРАЖДАНСКОЙ АВИАЦИИ

Кафедра прикладной математики

Домашнее задание защищено:

_____ (дата)

_____ (оценка)

Преподаватель: _____
(Фамилия, инициалы)

Дата: _____

Подпись: _____

Комплексное домашнее задание № 2
(Отчет)

Разработка алгоритма решения конкретной прикладной задачи и его
программной реализации в конкретной среде программирования на
конкретном языке программирования

Вариант № ____

Работу выполнил:
Студент (ка) группы _____

_____ (Фамилия, инициалы)

Дата: _____

Подпись: _____

Москва – 201_ г.

ПРИЛОЖЕНИЕ 2.

Примерный перечень вариантов заданий для КДЗ № 2

Вариант 1.

В старояпонском календаре был принят **60-летний цикл**, состоящий из **пяти 12-летних подциклов**, соответствующих последовательным цветам: зеленому, красному, желтому, белому и черному. Внутри каждого подцикла годы носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. (1948 год - год зеленой крысы - начало цикла). Определить по номеру **n** - год и вывести его название по старояпонскому календарю.

Исходная информация вводится с клавиатуры.

Вариант 2.

Астрологи делят год на 12 периодов по знакам Зодиака:

- 20.01-18.02 – **Водолей**; 19.02-20.03 - Рыбы; 21.03-19.04 - **Овен**;
- 20.04-20.05 - **Телец**; 21.05-21.06 - **Близнецы**; 22.06-22.07 - **Рак**;
- 23.07-22.08 - **Лев**; 23.08-22.09 - **Дева**; 23.09-22.10 - **Весы**;
- 23.10-22.11 - **Скорпион**; 23.11-21.12 - **Стрелец**; 22.12-19.01 - **Козерог**.

Для вводимой даты года вывести название соответствующего знака Зодиака.

Исходная информация вводится с клавиатуры.

Вариант 3.

Сформировать и вывести таблицу умножения в **шестнадцатеричной системе счисления**. Вывод осуществить в виде таблицы, на пересечении **i-ой строки и j-го столбца** которой располагается результат перемножения 16-ричных цифр “i” и “j”.

Исходная информация вводится с клавиатуры.

Вариант 4.

Дано натуральное число **N**, выражающее **цену товара в копейках**. ($N < 1000$). Вывести **цену товара в рублях и копейках** с использованием слов “рубли” и “копейки” (с учетом падежа и числа - единственного или множественного).

Исходная информация вводится с клавиатуры.

Вариант 5.

Дано натуральное число **N** ($20 < N < 1000$). Вывести **словесную запись этого числа** с учетом указанного падежа: (именительного, родительного, дательного, винительного, творительного, предложного). (Например: число - 147, родительный падеж: “ста сорока семи”).

Исходная информация вводится с клавиатуры.

Вариант 6.

Сформировать по числовому заданию **даты (число, месяц)** ее словесную форму с учетом падежей (именительного, родительного, дательного, винительного, творительного, предложного). (Например: дата - 21.12, родительный падеж: "двадцать первого декабря").

Исходная информация вводится с клавиатуры.

Вариант 7.

Сформировать по числовому заданию **времени (часы, минуты)** его словесную форму (в именительном падеже). (Например: время - 21:12 – “двадцать один час двенадцать минут”).

Исходная информация вводится с клавиатуры.

Вариант 8.

Рассортировать слова **исходной фразы a\$** по алфавиту.

Исходная информация (слова исходной фразы) вводится с клавиатуры.

Вариант 9.

Вывести **список полей (в формате – “буква-цифра”)** шахматной доски, достижимых из заданного поля **a\$** двумя ходами шахматного коня.

Исходная информация вводится с клавиатуры.

Вариант 10.

Дан **перечень дат (формат – “21 сентября 1982 г.”)**. Рассортировать эти даты в порядке возрастания.

Исходная информация (даты) вводится с клавиатуры.

Вариант 11.

По заданному **слову a\$** сформировать и вывести слова, буквы в которых являются всевозможными перестановками букв исходного слова.

Исходная информация (исходное слово) вводится с клавиатуры.

Вариант 12.

Написать программу, реализующую **операции (“+”, “-”, “*”, “/”) с комплексными числами**. Исходные числа и результат имеют форму: **"a + bi"**.

Исходная информация (исходные числа) вводится с клавиатуры.

Вариант 13.

Рассортировать **слова исходного перечня слов** одинаковой длины **a\$** по убыванию суммы их **ASCII-кодов**.

Исходная информация (исходные слова, перечень слов) вводится с клавиатуры.

Вариант 14.

Написать программу **алгебраического сложения десятичных чисел с плавающей запятой**. Исходные числа и результат должны иметь следующую форму: **“0.nnnnn*10^mmm”**.

Исходная информация (исходные числа) вводится с клавиатуры.

Вариант 15.

Из **исходной строки символов a\$** выбрать и распечатать наиболее длинные последовательности символов, у которых **ASCII-коды последовательно возрастают и убывают**.

Исходная информация вводится с клавиатуры.

Вариант 16.

В закрытом ящике находится **N белых и M черных шаров**. По исходному перечню вынутых шаров (заданному в форме: **“белый, черный, ...”**):

- рассчитать и вывести **вероятности появления очередного белого и черного шаров;**
- вывести также (**словом**) наиболее вероятный цвет следующего шара.

Исходная информация вводится с клавиатуры.

Вариант 17.

Рассортировать **символы исходной фразы а\$** по алфавиту.

Исходная информация вводится с клавиатуры.

Вариант 18.

По **введенной сумме** (в форме **“0.nn руб.”**) вывести в **словесной форме наименьшую (по числу монет) совокупность монет с их достоинствами** (одна копейка, пять копеек, десять копеек, пятьдесят копеек) **в убывающем порядке** (например: *введено – “0.23 руб.”, результат – “десять коп. + десять коп. + три коп.”*), **необходимую для заданной (введенной) суммы.**

Исходная информация вводится с клавиатуры.

Вариант 19.

Написать программу **умножения и деления десятичных чисел с плавающей запятой.** Исходные числа и результат должны иметь следующую форму: **“0.nnnpp*10^mmm”**.

Исходная информация (исходные числа) вводится с клавиатуры.

Вариант 20.

Проанализировать некоторое **символьное выражение а\$** с входящими в него **скобками** (типа: **“(“,”)”, “[“,”]”, “{“,”}”**) на **правильность расстановки скобок** (возможно, вложенных). Вывести ответ в форме: “верно/неверно”.

Исходная информация (символьное выражение а\$) вводится с клавиатуры.

Вариант 21.

По **введенной совокупности сочетаний символов, разделенных знаками “,”**, составить и вывести отдельные описания слов: **“ЦЕЛЫЕ ЧИСЛА - четные, нечетные”**; **“ДРОБНЫЕ ЧИСЛА - по числу знаков после десятичной точки”**; **“СЛОВА - по числу букв”**. Вывести комментарии по всем сочетаниям с учетом падежей (именительного, родительного, дательного, винительного, творительного, предложного).

Исходная информация вводится с клавиатуры.

Вариант 22.

Рассортировать слова **исходной фразы а\$** по алфавиту **в обратном порядке.**

Исходная информация вводится с клавиатуры.

Вариант 23.

Из **исходного текста а\$** выделить и распечатать все прилагательные.

Исходная информация вводится с клавиатуры.

Вариант 24.

По заданной фразе а\$ сформировать и вывести фразы, слова в которых являются всевозможными перестановками слов исходной фразы.

Исходная информация вводится с клавиатуры.

Вариант 25.

Написать программу распечатки результатов раздачи игральнх карт из колоды (по шесть карт трем игрокам) с указанием номиналов и мастей карт (например: “валет червей”).

Исходная информация вводится с клавиатуры.

Вариант 26.

Из заданной в виде строки а\$ совокупности чисел с дробной частью (разделитель целой и дробной части - точка, разделитель чисел - пробел) определить сумму целых частей чисел и вывести в текстовой форме.

Исходная информация вводится с клавиатуры.

Вариант 27.

Рассортировать символы исходной фразы а\$ по алфавиту в обратном порядке.

Исходная информация вводится с клавиатуры.

Вариант 28.

Написать программу сложения и умножения восьмеричных чисел с плавающей запятой. Исходные числа и результат должны иметь следующую форму: “0.nnnnl*8^mmm”.

Исходная информация (исходные числа) вводится с клавиатуры.

Вариант 29.

Рассортировать слова исходной фразы а\$ по алфавиту (по заданному номеру N буквы в слове).

Исходная информация (исходная фраза и заданный номер) вводится с клавиатуры.

Вариант 30.

Составить программу автоматического переноса слов.

Исходная информация вводится с клавиатуры.

Вариант 31.

Из исходного текста а\$ выделить и распечатать все существительные.

Исходная информация вводится с клавиатуры.

Вариант 32.

По N островам архипелага составлена годовая таблица среднемесячных температур. Найти такие острова, на которых среднемесячная температура не опускалась ниже M градусов, а также - какой месяц на каком острове был самым холодным. Вывести результаты анализа.

Исходная информация вводится с клавиатуры.

Вариант 33.

Пусть 1 января некоторого (не високосного) года - понедельник.

Вывести *все простые числа* в пределах от M до N (M и N - номера дней в году, $1 \leq M < N \leq 365$) и соответствующие им числа месяца, месяцы и дни недели. Также вывести те месяцы, 7-й день которых приходится на вторник.

Исходная информация вводится с клавиатуры.

Вариант 34.

Имеется таблица результатов шахматного турнира с N участниками. (Выигрыш - 1 очко, ничья - 0.5 очка, проигрыш - 0 очков). Вывести список фамилий участников в порядке убывания набранных ими очков, сопроводив его указанием для каждого участника суммы очков, занятого места и числа побед.

Исходная информация вводится с клавиатуры.

Вариант 35.

Дана таблица выпуска заводом N типов деталей по месяцам года. Из деталей собираются изделия. На одно (1) изделие идет M_i деталей i -го типа ($1 \leq i \leq N$). Вывести список числа выпускаемых изделий по месяцам (с учетом использования остатков от предыдущих месяцев); список остатков деталей по окончании года.

Исходная информация вводится с клавиатуры.

Вариант 36.

Дано расписание занятий в школе на неделю (6 дней, 6 уроков в день, N предметов). Вывести список предметов с указанием дней недели и числа уроков по дням в соответствии с расписанием. Вывести также названия предметов, количество уроков по которым больше заданного числа K . Выводимые данные отсортировать по алфавиту названий предметов.

Исходная информация вводится с клавиатуры.

Вариант 37.

Даны результаты бега N спортсменов в конькобежном многоборье на 4-х стандартных дистанциях: 500 м, 1000 м, 5000 м, 10000 м. Результаты (времена на дистанциях) должны вводиться реальные (сек.; мин., сек.; мин., сек.; мин., сек.). Сформировать таблицу баллов по дистанциям и суммарные баллы всех участников, приводя времена к 500-метровой дистанции (выражая все в сек.), число баллов пропорционально времени в секундах. Вывести таблицу, в которой участники соревнований отсортированы по местам, занятым ими (в формате: фамилия, баллы по дистанциям, сумма баллов, место).

Исходная информация вводится с клавиатуры.

Вариант 38.

Дан список N сотрудников с указанием фамилии, даты рождения, стажа работы и зарплаты. Вывести: список ровесников и список сотрудников со стажем, большим заданного числа K (с указанием всех данных по каждому сотруднику).

Исходная информация вводится с клавиатуры.

Вариант 39.

Дана таблица выпуска заводом N типов изделий по месяцам года;

список плановых заданий по каждому изделию ежемесячно; стоимость единицы каждого изделия. Вывести результаты выполнения плана (в %) по каждому изделию ежемесячно, поквартально и за год. Вывести также выполнение плана за год по стоимости произведенных изделий.

Исходная информация вводится с клавиатуры.

Вариант 40.

Дана таблица N учеников класса с указанием фамилии, имени, роста и веса. Дана также таблица дружащих групп учеников в классе (типа "да" - "нет"). Вывести (в полном формате: фамилия, имя, рост, вес) списки учеников, с которыми дружат самый высокий ученик и ученик с заданной фамилией.

Исходная информация вводится с клавиатуры.

Вариант 41.

Даны результаты квалификационных прыжков в длину N спортсменов (6 попыток). Вывести отдельно таблицы спортсменов, прошедших и не прошедших квалификационный рубеж (K см) в формате: фамилия, результаты (не засчитанные попытки: "-."). Вывести также абсолютно лучший результат и фамилию прыгуна.

Исходная информация вводится с клавиатуры.

Вариант 42.

Дана таблица начальных стоимостей 1 кг продуктов (N типов); в ней для каждого продукта указаны числа дней, после которых сорт его (высший, 1-й, 2-й, 3-й, "некондиция") снижается. Учитывая, что при снижении сортности продукта его стоимость падает на 20% , вывести цену, уплачиваемую покупателем в j -ый день при покупке им K килограмм (K_i кг) каждого i -го продукта ($1 \leq i \leq N$).

Исходная информация вводится с клавиатуры.

Вариант 43.

Дана таблица сдачи N студентами K экзаменов. Вывести список дисциплин с указанием средних баллов всех студентов по ним (по убыванию средних баллов), а также баллы по предметам и средние баллы лучшего и худшего из сдававших экзамены студентов.

Исходная информация вводится с клавиатуры.

Вариант 44.

Имеются сведения о багаже каждого из N пассажиров (фамилия, число мест, вес каждого места). Определить и вывести фамилии пассажиров, общий вес багажа которых превышает K килограмм, а также полные сведения (фамилия, число мест, вес каждого места) о пассажирах, багаж которых имеет максимальный и минимальный вес.

Исходная информация вводится с клавиатуры.

Вариант 45.

Даны результаты судейства N судей по M танцевальным парам фигуристов: техника и артистичность - баллы, места. С учетом отбрасывания для каждой пары низшей и высшей оценки вывести таблицу результатов, в которой танцевальные пары отсортированы по занятым

местам, в формате: **фамилии пар, сумма баллов, сумма мест** (отдельно по технике, отдельно по артистичности, отдельно общие). (Необходимо учитывать, что сумма мест имеет приоритет перед набранными баллами).

Исходная информация вводится с клавиатуры.

Вариант 46.

Дан список N сотрудников с указанием фамилии, точной даты рождения, стажа работы и заработной платы. **Вывести: список сотрудников, получающих заработанную плату выше средней по коллективу, упорядоченный по убыванию заработной платы** (в полном формате: с указанием всех данных (фамилия, точная дата рождения, стаж работы, заработная плата) по каждому сотруднику).

Исходная информация вводится с клавиатуры.

Вариант 47.

Дана **информация по видам товаров, экспортируемых в N стран по K годам** (в единицах стоимости). **Определить и вывести данные об общей стоимости товаров, экспортируемых в i-ом году в j-ую страну, а также список стран, стоимость суммарного экспорта в которые за все годы превысила L единиц.**

Исходная информация вводится с клавиатуры.

Вариант 48.

Дана **итоговая таблица футбольного чемпионата**, в котором **участвовало N команд**. Все элементы таблицы, принадлежащие главной диагонали, равны нулю, а другие элементы равны: 0, 1, 2 (т.е. числу очков, набранных в игре: 0 - поражение, 1 - ничья, 2 - победа). **Вывести число команд, имеющих больше побед, чем поражений, список команд с суммами набранных очков в порядке занятых мест по результатам чемпионата.**

Исходная информация вводится с клавиатуры.

Вариант 49.

Получены **результаты выборов в городскую Думу: по каждому из K округов известно число голосовавших за представителей каждой из M партий**, а также известно число не участвовавших в голосовании. **Выборы в округе считаются состоявшимися, если в голосовании приняло участие не менее P процентов избирателей. Определить, в каких округах был высший процент голосовавших избирателей и номера партий, получивших наибольшее число голосов в округах, где выборы считаются состоявшимися.**

Исходная информация вводится с клавиатуры.

Вариант 50.

Имеются **сведения об ассортименте N детских игрушек: название, цена, границы возраста детей**, для которых предназначена игрушка. **Определить и вывести: список игрушек, цена которых L лежит в пределах $a < L < b$; перечень названий и цен игрушек, пригодных для детей K-летнего возраста.**

Исходная информация вводится с клавиатуры.

Вариант 51.

Дан **список N сотрудников с указанием фамилии, точной даты**

рождения, стаж работы и заработной платы. Вывести: **список сотрудников**, возраст которых не менее K лет и не превышает L лет, упорядоченный по алфавиту фамилий (в полном формате, с указанием всех данных (фамилия, точная дата рождения, стаж работы, заработная плата) по каждому сотруднику).

Исходная информация вводится с клавиатуры.