

67639

89-4  
3164 а

ГОСУДАРСТВЕННЫЙ  
КОМИТЕТ СССР  
ПО ВЫЧИСЛИТЕЛЬНОЙ  
ТЕХНИКЕ И  
ИНФОРМАТИКЕ

ВСЕСОЮЗНЫЙ МЕЖОТРАСЛЕВОЙ  
НАУЧНО-УЧЕБНЫЙ ЦЕНТР  
ПО ВЫЧИСЛИТЕЛЬНОЙ  
ТЕХНИКЕ И  
ИНФОРМАТИКЕ

**ОСНОВЫ ИНФОРМАТИКИ  
И ВЫЧИСЛИТЕЛЬНОЙ  
ТЕХНИКИ ДЛЯ ПЭВМ  
«АГАТ»**

1988

ПРАКТИКУМ

89-4  
3164 а

ГОСУДАРСТВЕННЫЙ КОМИТЕТ СССР  
ПО ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ  
И ИНФОРМАТИКЕ

ВСЕСОЮЗНЫЙ МЕЖОТРАСЛЕВОЙ  
НАУЧНО-УЧЕБНЫЙ ЦЕНТР  
ПО ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ  
И ИНФОРМАТИКЕ

Г.Н. Волкова

ОСНОВЫ ИНФОРМАТИКИ  
И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ  
ДЛЯ ПЭЕМ "АГАТ"  
(для учащихся 10-х классов)

Москва 1988

Практикум

## ОБЩИЕ ПОЛОЖЕНИЯ

Преподавание курса "Основы информатики и вычислительной техники" введено во всех средних общеобразовательных школах в соответствии с принятым постановлением по обеспечению компьютерной грамотности учащихся средних учебных заведений и широкого внедрения электронно-вычислительной техники в учебный процесс в 1985-86 учебном году.

В соответствии с программой преподавание курса осуществляется в 9-х классах средних общеобразовательных школ в объеме 34 ч., т.е. один час в неделю. В 10-х классах в зависимости от возможности организовать практическую работу школьников на ЭМ обучение по этому курсу должно вестись в следующем объеме: в школах, имеющих вычислительную технику или возможность организовать систематические занятия в вычислительных центрах, - 68 ч, т.е. 2 ч в неделю; в школах, не имеющих таких возможностей, - 34 ч, т.е. один час в неделю.

Настоящий практикум предназначен для проведения занятий с учащимися средних учебных заведений по курсу "Основы информатики и вычислительной техники" в учебных классах (кабинетах вычислительной техники) на базе персональной ЭМ "Агат".

Материал раздела "Практические занятия" может быть использован в качестве раздаточного для учащихся на время выполнения практической работы, что повышает самостоятельное участие школьников в работе и облегчает труд преподавателя.

Практикум составлен в соответствии с действующей в настоящее время Программой курса для средних учебных заведений и рекомендаций органов народного образования.

Целью практических занятий является:

1. Закрепление полученных теоретических знаний:
  - средств и методов описания алгоритмов;
  - основных типов алгоритмов (линейный, разветвляющийся, циклический);

88-67639

- программ как формы представления алгоритма для ЭВМ;
- основ программирования на языке Бейсик-Агат;
- принципов действия и устройств ЭВМ на примере персональной ЭВМ "Агат";
- применения и роли компьютеров в различных отраслях деятельности человека.

2. Приобретение учащимися практических навыков общения с ЭВМ на примере персональной ЭВМ "Агат":

- умение подготовить ЭВМ к работе;
- освоение клавиатуры;
- умение вводить в память ЭВМ числа и переменные;
- умение корректировать введенные в ЭВМ данные;
- умение вводить и отлаживать программу на ЭВМ;
- умение подключать к ЭВМ внешние устройства, управлять работой ЭВМ.

#### МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

В существующих в настоящее время методических рекомендациях по курсу содержатся только ориентировочные методические схемы. Применительно к конкретным условиям имеющуюся схему наполняют определенным содержанием, придавая ей соответствующую форму.

Содержание практических занятий данного практикума определено теоретическим материалом, изучаемым на занятиях, и возможностями персональной ЭВМ "Агат".

Все практические работы оформлены по единому плану и включают в себя следующие пункты:

- название темы работы;
- указание цели работы;
- продолжительность выполнения;
- краткие теоретические сведения;
- контрольные вопросы;
- порядок выполнения работы;
- требования к отчету;
- задания к работе (с вариантами);
- пример выполнения задания.

При изложении основных теоретических сведений во всех работах подчеркивается связь между конструкциями школьного алгоритмического языка и языка конкретной ПЭВМ Бейсик-Агат.

Контрольные вопросы могут быть использованы для фронтального опроса. Ответы на контрольные вопросы, не являющиеся прямым отражением в теоретической части, приведены в Приложении I. Теоретические сведения, помещаемые в практической работе, могут быть использованы для самоподготовки учащихся к выполнению данной практической работы и как справочный материал в случае возникновения трудностей в процессе ее выполнения.

Задания к практическим работам целесообразно выдавать с опережением, чтобы учащиеся могли хорошо продумать ход решения задачи, построить алгоритм, написать программу, а время, отведенное на практические занятия, использовать для отладки и выполнения программы на ПЭВМ.

Выдавать задания к практическим работам можно в качестве домашних работ при изучении теоретической части курса.

При выполнении заданий учащиеся обязаны составить требуемую программу на школьном алгоритмическом языке и на языке Бейсик-Агат. Необходимость составления алгоритма решаемой задачи в виде блок-схемы определяется преподавателем.

Практические занятия на ПЭВМ для учащихся 9-х и 10-х классов заканчиваются зачетными работами. Это самостоятельные работы учащихся, выполняемые под руководством преподавателя. Варианты зачетных работ могут быть даны учащимся после первого практического занятия с целью постепенного ее выполнения по мере закрепления теоретических знаний и приобретения навыков работы на ПЭВМ. Время, отведенное на зачетное занятие, должно использоваться для отладки, выполнения заданий на ПЭВМ, защиты зачетной работы у преподавателя.

Порядок выполнения работ может изменяться исходя из возможностей проведения занятий в кабинетах вычислительной техники и изучения теоретической части курса.

ПРАКТИЧЕСКИЕ ЗАНЯТИЯ  
ПРАКТИЧЕСКАЯ РАБОТА 1

Тема: Персональная ЭВМ.

Цель работы: Изучение правил техники безопасности при работе в кабинете вычислительной техники. Ознакомление с основными блоками ПЭВМ; правилами включения-выключения ПЭВМ и правилами ввода и редактирования информации.

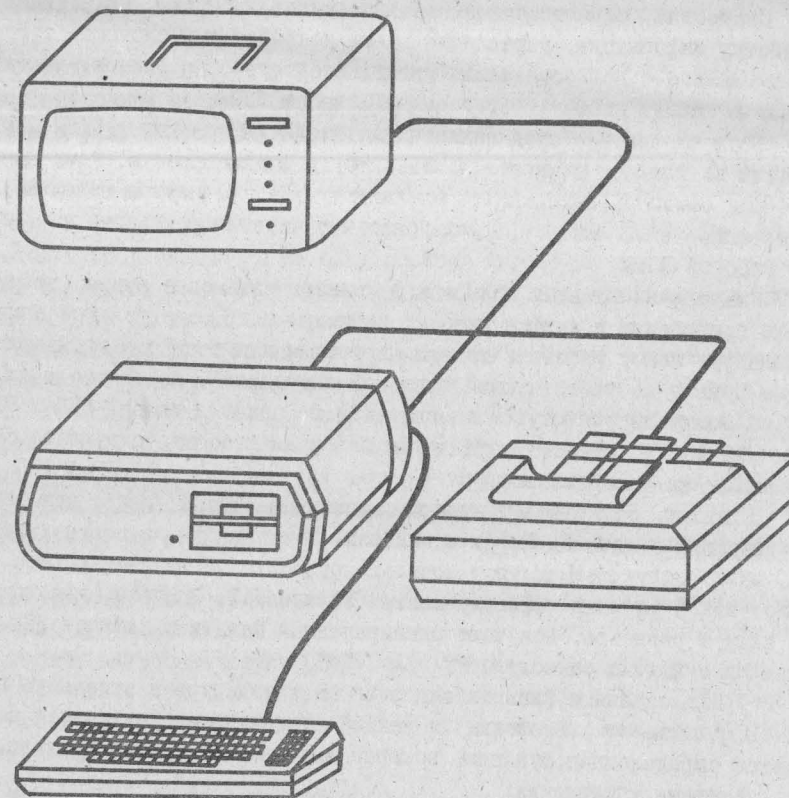
Продолжительность работы - 2 ч.

Теоретические сведения. Перед началом работы в кабинете вычислительной техники необходимо ознакомиться с правилами техники безопасности, соблюдение которых обеспечит безопасную и стабильную работу аппаратуры. Памятка с правилами находится на каждом рабочем месте. Текст ее приведен в Приложении 2.

Персональный компьютер предназначен для автоматической обработки информации и обеспечивает ввод программы и данных с клавиатуры или магнитного диска, редактирование программ и данных, работу в диалоговом режиме, вывод программы и результатов ее работы на экран видеоконтрольного устройства (ВКУ), печатающее устройство или магнитный диск.

Чтобы выделить из огромного количества выпускаемых ЭВМ класс персональных, определим следующие требования, которым они должны удовлетворять:

- настольное исполнение;
- возможность работы в помещениях с бытовым электропитанием, не имеющих оборудования для поддержания постоянной температуры;
- широкая номенклатура внешних устройств, подключаемых к ПЭВМ;
- работа в символично-графическом диалоге;
- использование версий языков Бейсик, Паскаль, КОБОЛ, ФОРТРАН;
- возможность подключения по каналам связи к любым ЭВМ;
- повышенная надежность исполнения.



Объединение составных частей ПЭВМ "Агат"

Персональная ЭВМ "Агат" состоит из (рис.):

- системного блока с встроенным накопителем на гибких магнитных дисках (НГМД);
- блока клавиатуры;
- видеоконтрольного устройства;
- мозаично-печатающего устройства.

Системный блок обеспечивает выполнение программ, хранение и обработку информации, управление всеми устройствами системы, организацию диалога с пользователем. Он является основным блоком, объединяющим в единую систему все составные части ПЭВМ.

Блок клавиатуры предназначен для ввода информации в ПЭВМ. При нажатии на клавишу замыкаются контакты, и в электронную схему попадает код символа, нанесенного на клавишу. На клавиатуре выделено три зоны: символьные клавиши, функциональные клавиши и клавиши управления работой ПЭВМ.

Символьные клавиши содержат русские и латинские буквы, цифры, знаки препинания и арифметических действий и др. специальные символы. Клавиатура может работать только с русским или только с латинским набором букв с соответствующей световой индикацией. Из одного режима в другой можно перейти путем одновременного нажатия клавиш REG и RUS, либо REG и LAT, что подтверждается загоранием соответствующего светового указателя на блоке клавиатуры.

Клавиши, содержащие цифры и специальные знаки, имеют два регистра: верхний и нижний. Цифры и символы, находящиеся на верхнем регистре, можно получить нажатием соответствующей клавиши независимо от того, какой регистр зафиксирован - RUS или LAT. Символы, находящиеся на нижнем регистре, получают одновременным нажатием клавиши REG и клавиши с нужным символом.

Функциональные клавиши выделены на клавиатуре в отдельную группу. В отличие от символьных за каждой функциональной клавишей закрепляется определенная функция, которая выполняется при ее нажатии.

Клавиши управления:

СБР	- сброс	УПР	- управление
	- перевод строки		- управление курсором вниз
	- повторение		- управление курсором вверх
	- регистр		- управление курсором вправо
	- редактирование		- управление курсором влево

Назначение и использование клавиш управления приведены в Приложении 3.

Видеоконтрольное устройство цветного отображения обеспечивает диалоговый режим работы. Экран ВКУ предназначен для отображения вводимой и выводимой символьной и графической информации. Каждый символ располагается в определенном столбце и определенной строке (позиции) экрана. Одна из позиций отмечена знаком - мигающим курсором. Символ, вводимый с клавиатуры, отображается в позиции, отмеченной курсором. При этом курсор смещается на одну позицию вправо по текущей строке до тех пор, пока не дойдет до ее конца. Достигнув конца строки, курсор переходит на начало следующей строки.

Если вводимая информация не помещается на одной строке экрана, то вводимая строка занимает место нижней, а все ранее введенные смещаются на одну строку вверх, при этом самая верхняя строка исчезает с экрана. Курсор перемещается по экрану ВКУ с помощью клавиш , , (их нажатие вызывает смещение курсора на одну позицию вправо, влево, вверх и вниз соответственно).

Мозаично-печатающее устройство, или принтер, не является обязательным устройством и может быть отключено от ПЭВМ без нарушения нормальной работы. Принтер предназначен для получения твердой копии информации, хранимой в ПЭВМ.

Гибкие магнитные диски (ГМД), или дискеты, предназначены для долговременного хранения большого количества информации. ГМД представляет собой пластиковый диск с магнитным покрытием, помещенный в плотный бумажный пакет.

При работе с ГМД запрещается:

- касаться магнитного покрытия диска;
- подносить к ГМД близко магниты;
- сгибать и складывать диск, допускать механические повреждения пакета и самого диска.

Односторонние магнитные диски вставляются в дисковод маркировочной этикеткой вверх, прорезью для считывающей головки вперед.

Порядок включения-выключения ПЭВМ "Agat":

- 1) включаем ВКУ, нажимая на передней панели клавишу;
- 2) включаем системный блок тумблером на задней панели при закрытой крышке дисковода;

3) вставляем ГМД в НГМД, закрываем крышку, и начинается загрузка дисковой операционной системы, о чем сообщает светящийся индикатор на НГМД. По завершении загрузки индикатор гаснет, и на экране ВКУ появляются знак правой квадратной скобки и мигающий курсор. ПЭВМ готова к работе.

Выключается ПЭВМ в обратном порядке: сначала системный блок, а затем ВКУ.

Основные операции редактирования. Для замены одного символа другим курсор подводит к заменяемому символу и нажимают соответствующую клавишу. Удаление (стирание) символа в позиции курсора выполняется нажатием клавиши "Пробел" (самая длинная клавиша, находящаяся в нижнем ряду блока клавиатуры).

Чтобы внести изменения в строку программы, находящуюся в памяти ЭВМ, нужно указать номер этой строки, а затем правильный текст. Строка, имеющая указанный номер, заменится на вновь введенную.

Если изменения в строке незначительны, то целесообразно вводить правильный вариант не полной строки программы, а откорректировать имеющуюся строку. Для этого текст неверной строки должен быть на экране ВКУ. После правой квадратной скобки вводится номер редактируемой строки, и курсор с помощью управляющих стрелок устанавливается на начало редактируемой строки. Символы, которые остаются неизменными, проходят клавишей  $\leftarrow$ , неверные символы заменяются новыми нажатием соответствующей клавиши. Строку нужно обязательно пройти до конца и нажать клавишу  $\downarrow$ , чтобы измененная строка записалась в память ПЭВМ.

Если нужно удалить лишние символы в середине строки, курсор устанавливается под первым удаляемым символом и нажимают клавишу  $\text{РЕД}$ . В позиции курсора появляется мигающая точка. Далее клавишей  $\rightarrow$  проходят все удаляемые символы и, выведя мигающую точку на символ, подлежащий удалению, нажимают клавишу  $\text{РЕД}$ . Оставшиеся символы проходят курсором до конца с помощью клавиши  $\rightarrow$  или заменяют новыми. Исправленная строка запоминается в оперативной памяти ПЭВМ при нажатии клавиши  $\downarrow$ .

Если нужно отбросить конец строки, то курсором проходят оставшиеся в строке символы и нажимают клавишу  $\downarrow$ .

Если строку нужно раздвинуть для вставки недостающих символов, то курсор устанавливается на позицию в редактируемой строке, в которой

должен быть вставлен первый недостающий символ. Потом клавишей  $\uparrow$  или  $\downarrow$  курсор перемещают на чистое поле экрана и вводят недостающие символы. Затем курсор возвращают на редактируемую строку клавишей  $\uparrow$  или  $\downarrow$ . Устанавливают режим редактирования, нажав клавишу  $\text{РЕД}$ . Вместо курсора появляется мигающая точка. Возвращают курсор клавишей  $\leftarrow$  на первый символ, следующий за вставкой. Выходят из режима редактирования, нажав клавишу  $\text{РЕД}$ . Курсор перемещают до конца редактируемой строки и нажимают клавишу  $\downarrow$ . ПЭВМ запоминает измененную строку программы.

#### Контрольные вопросы

1. Перечислить правила техники безопасности при работе в кабинете вычислительной техники.
2. Каково назначение основных блоков ПЭВМ?
3. Указать последовательность действий при включении-выключении ПЭВМ.
4. Определить последовательность нажатия клавиш для ввода текста

$$\text{ДИСКРИМИНАНТ} = B \wedge 2 - 4 * A * C$$

если зафиксирован латинский регистр.

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задания 1, 2, 3;
- оформить отчет.

Отчет по практической работе 1 должен содержать:

- название и цель работы;
- перечень основных блоков ПЭВМ "Агат";
- описание порядка включения-выключения ПЭВМ;
- ответ на вопрос: в какой последовательности должны быть нажаты клавиши для ввода текста

$$\text{ПОЛУПЕРИМЕТР} = (A + B + C)/2$$

если зафиксирован латинский регистр?

Задание 1 выполняется с использованием обучающей программы

"Знакомство с клавиатурой", которая выдает упражнения и контролирует правильность выполнения задания, а также оценивает знания учащихся.

Программа начинает работать при последовательном нажатии клавиш

R U N  $\downarrow$ .

Задание 2. С помощью символа "." (точка) нарисуйте на экране ВКУ домик следующего вида:



Задание 3.

1. Введите предложение

10. " ЭВМ РЕДАКТИРУЕТ ТЕКСТ "

2. Замените слово:

- а) "ЭВМ" словом "ОНА";
- б) "ТЕКСТ" словом "ТЕКСТЫ";
- в) "РЕДАКТИРУЕТ" словом "ИСПРАВЛЯЕТ".

Пример выполнения задания.

\* \* \*ЗНАКОМСТВО С КЛАВИАТУРОЙ\* \* \*

НАБЕРИТЕ: 1987

?1987

ПРАВИЛЬНО!

НАБЕРИТЕ: 0.876

?0,876

НЕПРАВИЛЬНО! ПРОВЕРЬТЕ, КАКИМ АЛФАВИТОМ И РЕГИСТРОМ ВЫ ПОЛЬЗУЕТЕСЬ!

НАБЕРИТЕ: 0.876

?-

и т.д.

После выполнения Вами всех упражнений обучающая программа выводит на экран ВКУ одно из трех сообщений:

"ХОРОШО ! (ОТЛИЧНО!)"

ВЫ ГОТОВЫ К РАБОТЕ С ПРОСТЫМИ ПРОГРАММАМИ НА ЯЗЫКЕ БЕЙСИК-АГАТ!"

"УДОВЛЕТВОРИТЕЛЬНО!"

ВАМ НЕОБХОДИМО ПОРАБОТАТЬ С ДАННОЙ ПРОГРАММОЙ ЕЩЕ РАЗ!"

"ПЛОХО, НО НЕ БЕЗНАДЕЖНО!  
БУДЬТЕ БОЛЕЕ ВНИМАТЕЛЬНЫ  
ПРИ РАБОТЕ С КОМПЬЮТЕРОМ!"

Если Вы получили оценку "отлично" или "хорошо", поднимите руку и сообщите об этом преподавателю, а затем выполняйте следующее задание.

Если Вы получили: оценку "удовлетворительно" или "плохо", то программа предложит Вам повторить все упражнения еще раз.

ПРАКТИЧЕСКАЯ РАБОТА 2

Тема: Общие сведения о языке Бейсик.

Цель работы: Ознакомление с типами констант и переменных, знаками арифметических действий, приобретение навыков работы на ПЭВМ "Агат" в режиме микрокалькулятора.

Продолжительность работы - 2 ч.

Теоретические сведения. Язык программирования Бейсик предназначен для реализации на ЭВМ алгоритмов решения числовых задач в режиме диалога.

Программа на языке Бейсик состоит из последовательности строк. Каждая строка начинается номером (или числовой меткой). Номера строк могут быть целые положительные числа от 0 до 9999. Номера строк указывают последовательность выполнения операторов и должны следовать в возрастающем порядке, но не обязательно подряд. Однако обычно для удобства последующих корректировок строки нумеруются с интервалом 10.

В каждой строке можно записать одну или несколько команд. Если в одной строке записано несколько команд, их разделяют двоеточием, например:

100 A=2 : B=3 : C=A+B

Одна строка программы на экране ВКУ может занимать более одной физической строки. Физическая строка на экране содержит 32 символа, а максимально допустимая строка программы - 255 символов.

Строки программы можно вводить в произвольном порядке (например, сначала сотую, а потом десятую), так как при размещении в памяти они автоматически упорядочиваются.

В языке Бейсик нет таких строгих правил оформления программы, как в алгоритмическом языке. Но в первых строках программы в коммен-



тарихх желательнo дать короткое объяснение сущности алгоритма и переменных. Для этого используется оператор REMark (заметка):

REM <строка символов >

Встретив оператор REM, интерпретатор Бейсика игнорирует все следующие за этим ключевым словом символы до конца физической строки экрана и переходит к выполнению следующей строки программы.

Как и в алгоритмическом языке, в языке Бейсик величины делятся на переменные и константы. Каждая величина может быть целого, вещественного или литерного типа (табл. 1).






Т а б л и ц а 1

Тип величины	Диапазон использования	Пример записи	Определение
Вещественные (вещ)*	$10^{-38}$ $+10^{+38}$	50.3 -148.12 42861 0.85 6.02E23 1.6E-19	Набор из одной или нескольких десятичных цифр, положительный или отрицательный, целый или дробный. Отделение целой части от дробной обозначается точкой
Целые (цел, нат)	-32768% $+$ $+32767\%$	18% -375% 0%	Набор десятичных цифр, оканчивающихся знаком %
Строковые, или литерные, или текстовые (лит)	0 $+$ 255 символов	ПАРОХОД P=СУММЕ A+B A#BC SIRIUS MAR(V) A BC R20 385	Набор буквенно-цифровых или специальных символов. Имя строковой константы или переменной оканчивается знаком "X" (солнышко). Явно задаваемая строковая константа ограничивается с двух сторон двойными кавычками

\* В скобках указано соответствие типа обозначению в алгоритмическом языке.

На клавиатуре ПЭЕМ имеются клавиши, нажатие которых вызывает выполнение соответствующих арифметических операций (табл. 2).

Т а б л и ц а 2

Клавиша	Действие, выполняемое при нажатии клавиши
	Возведение в степень
	Умножение
	Деление
	Сложение
	Вычитание

В выражениях на языке Бейсик порядок выполнения действий может быть выражен табл. 3.

Т а б л и ц а 3

Порядок выполнения операций	Выполняемая операция
1	Действия в скобках
2	Вычисление стандартных встроенных функций SQR, SIN и др.
3	Возведение в степень
4	Одноместный минус, изменяющий знак стоящей за ним величины
5	Умножение, деление
6	Сложение, вычитание

Операции, имеющие равный приоритет, выполняются слева направо. При использовании скобок первым выполняется самое внутреннее выражение, затем внешнее по отношению к предыдущему и т.д. Другими словами, здесь действуют те же правила, что и в математике.

ПЭЕМ "Агат" может работать в режиме микрокалькулятора. Для того чтобы пользователь мог увидеть результат просчета заданного им арифметического выражения, перед самим выражением нужно дать ключевое слово PRINT (печатать) или знак "?".

### Контрольные вопросы

1. Как оформляются программы на языке Бейсик?
2. Какие типы величин используются в языке Бейсик?
3. Рассказать о выражениях на языке Бейсик и приоритете операций.
4. Каково назначение оператора REM?

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задание 1 или 2 по вариантам;
- оформить отчет.

Отчет по практической работе 2 должен содержать:

- название и цель работы;
- описание порядка выполнения действий в выражениях на языке Бейсик;
- запись общего вида и описание назначения оператора REM;
- условие задания 1 или 2, результаты выполнения.

Задание 1. Просчитать в режиме микрокалькулятора следующие выражения:

$$\begin{array}{l} (2 \cdot 3 + 4) : 13 \\ 37 \cdot (43 - 19) \\ 3, 14 \cdot 5 \cdot 6, 8 \end{array} \quad \begin{array}{l} \frac{2}{13 + 14,7} \cdot 8^2 \\ \sqrt{13 + (3 \cdot 16) : 2} \end{array}$$

Задание 2. Вычислить в режиме микрокалькулятора следующие выражения:

$$\begin{array}{l} 2^3 - \sqrt{4 + 13,8} \\ \frac{16,4^2 - 340}{13,4^2 : 64} \\ \frac{18 - 2^2}{(53 - 3^3 \cdot 13) : 5} \end{array} \quad \begin{array}{l} 836 : 4 + 562 \\ \sqrt{568 : 24 + 32} \end{array}$$

Пример выполнения задания. Вычислить в режиме микрокалькулятора следующие выражения:

$$(23,5 : 5 + 84) : \frac{84}{9} = ? (23,5 / 5 + 84) / (84 / 9) = 9,5 \overline{) 357143}$$

$$\frac{\sqrt{16+32^2}}{(184-56,8)} = ? (16 \wedge 0,5 + 32 \wedge 2) / (184 - 56,8 / 124) = 5,6 \overline{) 89986}$$

### ПРАКТИЧЕСКАЯ РАБОТА 3

Тема: Простейшие программы, реализующие линейный алгоритм на языке Бейсик, выполнение их на ПЭЕМ.

Цель работы: Знакомство с основными системными командами, командами языка Бейсик, реализующими линейный алгоритм, составление и отладка простейших программ на ЭЕМ.

Продолжительность работы - 2 ч.

Теоретические сведения. При работе на ПЭЕМ "Агат" пользователь должен знать основные системные команды, дающие возможность составлять, редактировать и работать с программами на языке Бейсик (табл.4).

Т а б л и ц а 4

Команда	Выполняемое действие
NEW	Очистить память
RUN	Запустить программу на выполнение
LIST	Вывести на ВКУ текст программы
DEL	Стереть строки программы из памяти

Команда RUN имеет два варианта записи и выполнения (табл. 5).

Т а б л и ц а 5

Запись	Выполняемое действие
RUN	Запуск программы на выполнение, начиная с наименьшего номера строки, встреченного в тексте программы
RUN N	Запуск программы на выполнение, начиная с номера строки N (например, RUN 100)

Команда NEW не имеет параметров и всегда записывается однозначно.

Команда LIST имеет пять вариантов записи и выполнения (табл.6).

Т а б л и ц а 6

Запись	Выполняемое действие
LIST	Вывод на экран ВКУ всего текста программы
LIST N1	Вывод на экран ВКУ строки программы с номером N1
LIST N1,N2	Вывод на экран ВКУ текста программы, начиная со строки с номером N1 по строку с номером N2
LIST N1,	Вывод на экран ВКУ текста программы, начиная со строки с номером N1 до конца текста
LIST, N2	Вывод на экран ВКУ текста программы сначала до строки с номером N2 включительно

Например, запись LIST 40, означает, что надо вывести на экран ВКУ текст программы, начиная со строки с номером 40 до конца.

Команда DEL всегда записывается с двумя параметрами, указанными через запятую. Но при этом от значения параметров зависит вариант выполнения (табл. 7).

Т а б л и ц а 7

Запись	Выполняемое действие
DEL N1,N2	Удаление из текста программы группы строк, начиная с номера N1 по номер N2
DEL N1,N1	Удаление из текста программы одной строки с номером N1

Возможен и более простой способ удаления одной строки без использования команды DEL. В этом случае после правой квадратной скобки указывается номер удаляемой строки и нажимается клавиша .

В составленной программе могут быть ошибки. В этом случае после команды RUN, запускающей программу на выполнение, выводятся сообщение об ошибке и текст ошибочной строки, в которой красным цветом выделяется ошибка. Перечень сообщений об ошибках приведен в Приложении 4.

Чтобы написать простейшую программу на языке Бейсик, реализующую линейный алгоритм, познакомимся с некоторыми командами.

В практической работе 2 мы уже рассмотрели оператор REM, которым обычно начинается любая программа. Данный оператор условно может быть аналогом ключевого слова алг, употребляемого в алгоритмическом языке, а аналогом ключевого слова кон в языке Бейсик является оператор END (конец). Выполняя его, ПЭВМ переходит из программного режима в командный и ожидает ввода пользователем следующего оператора, т.е. выполнение программы заканчивается.

Как и в алгоритмическом языке, в языке Бейсик есть оператор присваивания LET (давать). Это один из наиболее употребляемых операторов любого алгоритмического языка. Он единственный, распознаваемый интерпретатором по его форме, даже без названия.

Полная форма оператора присваивания LET:

$$\text{LET} \left\langle \begin{array}{l} \text{идентификатор} \\ \text{переменной или} \\ \text{элемента массива} \end{array} \right\rangle = \left\langle \begin{array}{l} \text{арифметическое} \\ \text{выражение} \end{array} \right\rangle$$

Сокращенная форма оператора присваивания опускает ключевое слово LET.

Идентификатор – это имя переменной, состоящее из последовательности букв и цифр. Первой в имени обязательно должна стоять буква латинского алфавита.

#### Контрольные вопросы

1. Каково назначение основных системных команд?
2. Каково назначение команд END и LET?
3. Как отреагирует ПЭВМ на команду DEL 110,110...?
4. Что произойдет, если будет дана команда LIST, 110?
5. Как отработает программа, если дана команда RUN/100?

Будут ли выполнены команды с номерами меньше 100?

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задание 1 или 2 по вариантам;
- оформить отчет.

Отчет по практической работе 3 должен содержать:

- название и цель работы;
- таблицу основных системных команд;
- запись общего вида и описание назначения операторов END и

LET;

- условие задания 1 или 2, текст составленной программы на алгоритмическом языке и на языке Бейсик.

Задание 1. Составить программу нахождения среднего арифметического шести чисел:  $a=18$ ,  $b=14,8$ ;  $c=2,2$ ;  $d=348$ ;  $e=0,365$ ;  $k=32$ .

Задание 2. Составить программу нахождения факториала числа  $8(8!)$ .

Пример выполнения задания. Составить программу нахождения

$D = \frac{A+B+C}{(A+B)^2}$ , если переменным заданы следующие значения:  $A=84$ ,  $B=362$ ,  $C=8$ .

Программа на алгоритмическом языке:

алг вычисление (вещ A, B, C, D)

арг A, B, C

рез D

нач

A:=84

B:=362

C:=8

D:=  $\frac{A+B+C}{(A+B)^2}$

кон

Программа на языке Бейсик:

10 REM ВЫЧИСЛЕНИЕ ПО ФОРМУЛЕ

20 A=84 : B=362 : C=8

30 D=(A+B+C)/(A+B)^2

40 END

## ПРАКТИЧЕСКАЯ РАБОТА 4

Тема: Операторы ввода и вывода в языке Бейсик. Выполнение операций ввода-вывода на ПЭМ.

Цель работы: Изучение команд ввода-вывода в языке Бейсик, их практическое применение в составляемых программах.

Продолжительность работы - 2 ч.

Теоретические сведения. Из предыдущей практической работы мы узнали, что программы, не имеющие операторов, позволяющих вводить исходные данные и выводить результат работы программы, примитивны и лишены наглядности. Работая со своими программами, Вы не увидите результата вычисления. Как же сделать программу более совершенной?

Для того чтобы ответить на этот вопрос, рассмотрим оператор вывода на экран ВКУ информации - PRINT (печатать).

Оператор имеет следующий вид:

PRINT <список вывода >

Список вывода может содержать строковые константы, переменные всех типов и элементы массивов. Если список вывода содержит не один элемент, а несколько, то они отделяются друг от друга запятыми или точкой с запятой. Эти разделители управляют размещением информации на экране.

Если разделителем является точка с запятой (;), то информация будет выводиться подряд, без пробелов. Когда разделитель ";" стоит за последним элементом списка вывода следующего оператора, т.е. печать будет происходить в ту же строку. В противном случае каждый следующий оператор PRINT осуществляет вывод с новой строки.

Разделитель запятая (,) задает печать в очередную зону экрана. Каждая зона кратна восьми разрядам. Стоящие подряд две запятые в списке вывода приведут к пропуску одной свободной зоны.

Ввод информации осуществляется с помощью оператора INPUT (ввод):

INPUT <список ввода >

При вводе нескольких величин их нужно перечислить через запятую. Ввод заканчивается при нажатии клавиши

В списке ввода могут быть указаны имена констант или переменных целого, вещественного или строкового типа. Как только начнет работать содержащийся в программе оператор INPUT, на экране ВКУ появятся

знак "?" и мигающий курсор. Это означает, что программа ожидает ввода с клавиатуры значений переменных.

Когда с программой работает не автор, а пользователь, ему трудно догадаться, ввода каких значений ожидает программа. Поэтому желательно использовать оператор ввода с запросом:

```
INPUT "<запрос>"; <список ввода >
```

В запросе содержится информация, которая дает пояснения о вводимых значениях, например:

```
50 INPUT "ВВЕДИТЕ ЗНАЧЕНИЯ ОСНОВАНИЯ  
И ВЫСОТЫ ТРЕУГОЛЬНИКА"; A, H
```

Если Вы введете меньшее количество значений, чем запрашивает оператор INPUT (в нашем примере одно вместо двух), то на экране ВКУ появляются два знака "?". Процессор просит Вас продолжить ввод до тех пор, пока количество введенных чисел не станет равным количеству переменных, перечисленных в операторе ввода.

Если Вы не хотите вводить новые значения всех переменных, перечисленных в списке ввода, то достаточно вместо ненужного значения набрать запятую. Так, если в нашем примере Вы наберете ?, 8 , то переменная A не изменится, т.е. будет иметь значение, присвоенное ей ранее, а переменная H примет значение 8.

Значения строковых переменных вводятся также. При этом заключать в кавчки их необязательно. Но ограждающие кавчки, если они есть, могут отстоять от крайних символов строки, и образующиеся при этом пробелы будут включены в значение вводимой строки. Таким образом можно вводить и пустую строку, т.е. ряд пробелов. Если же строка вводится без кавчек, то такая возможность отпадает.

В том случае, если вводимый текст должен содержать кавчки, двойные кавчки заменяются одинарными, например:

```
180 A X =КРЕЙСЕР _ 'АВРОРА'  
340 C X ="БРОНЕНОСЕЦ _ 'ПОТЕМКИН."
```

Знак одинарной кавчки находится на нижнем регистре клавиши, имеющей в верхнем регистре цифру 7.

Данные, необходимые для выполнения программы, можно записать с помощью оператора DATA (данные):

```
DATA <список данных >
```

Данные в списке данных перечисляются через запятую. Оператор DATA относится к числу невыполняемых. Он может стоять в любом месте

программы. Обращение к нему происходит лишь при выполнении оператора чтения READ (читать):

```
READ <список чтения >
```

Данные из списка данных оператора DATA считываются оператором READ без перерыва в работе программы. В программе допустимы несколько операторов DATA и READ.

При выполнении оператора READ в программе отыскивается первый оператор DATA и считаются перечисленные далее значения переменных. При этом первой переменной, стоящей за ключевым словом READ, присваивается первое значение, стоящее за ключевым словом DATA, второй переменной – второе значение и т.д.

В этом случае специальный указатель данных, настроенный сначала на первое значение в первом блоке DATA, перемещается к очередному его неиспользованному значению, например:

```
10 READ A, B, C  
20 READ D, E, F  
30 DATA 1,2,3,4,5,6
```

После выполнения этого фрагмента программы переменные примут следующие значения: A=1; B=2; C=3; D=4; E=5; F=6.

Программист по своему усмотрению может менять положение указателя, очищать его, устанавливая к началу первого блока данных с помощью оператора восстановления RESTORE (восстановить).

Если в рассмотренный выше пример добавить еще одну строку

```
15 RESTORE
```

то переменные примут другие значения: A=1; B=2; C=3; D=1; E=2; F=3.

Если оператор READ присвоил значения еще не всем своим переменным, а блок DATA, из которого эти значения считываются, исчерпан уже полностью, то в программе отыскивается и считывается следующий блок данных. Если же он не найден, то выдается сообщение об ошибке

```
NO DATA ERROR
```

означающее "мало данных" (см. Приложение 3).

Хотя операторы DATA могут стоять в любом месте программы, желательно ставить их рядом в конце программы. Так удобнее отлаживать программу и следить за правильностью считывания значений, перечисленных в блоке данных.

### Контрольные вопросы

1. Каково назначение операторов PRINT и INPUT?
2. Как будет выведена на экран информация, если программа имеет следующие операторы:  
30 V=3: T=40  
100 PRINT "ОНА ИДЕТ СО СКОРОСТЬЮ - " ;V, " ЗА ВРЕМЯ - ";T;  
130 PRINT " МИН"?
3. Для каких целей используется оператор ввода, содержащий запрос?
4. Каково назначение операторов DATA, READ, RESTORE?
5. Что произойдет, если количество переменных в операторе READ превышает количество данных в операторе DATA?

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задание 1;
- выполнить задание 2 или 3 по вариантам;
- оформить отчет.

Отчет по практической работе 4 должен содержать:

- название и цель работы;
- запись общего вида и описание назначения операторов ввода-вывода;
- условие задания 1, измененный текст программы на языке Бейсик, результат решения;
- условие задания 2 или 3, текст программы на алгоритмическом языке и на языке Бейсик, результат работы программы.

**Задание 1.** Используя оператор вывода, изменить текст программы на языке Бейсик, составленной Вами в предыдущей практической работе, так, чтобы результат ее работы выводился на экран ВКУ. Отладить программу.

**Задание 2.** Составить программу, реализующую алгоритм вычисления пропорциональной зависимости. Известные данные считывать из блока данных. Для отладки программы использовать следующую задачу.

Суточное потребление топлива на заводе составило до проведения рационализации 1,8 т; годовой расход на топливо составлял 3000 руб. После проведения рационализации суточное потребление снизилось до 1,5 т. Какую сумму расходов на топливо нужно запланировать на год?

**Задание 3.** Составить программу вычисления среднесуточной температуры воздуха, если измерения проводились каждые три часа. Данные измерений считывать из блока данных (цифры взять произвольные).

**Пример выполнения задания.** Составить программу вычисления значения

$$D = \frac{4 \cdot A - A \cdot B}{3 \cdot A \cdot C - 4 \cdot B \cdot C}$$

Переменные читать из блока данных.

Программа на алгоритмическом языке:

алг вычисление (вещ A, B, C, D)

арг A, B, C

рез D

нач

читать данные

$$D := \frac{4 \cdot A - A \cdot B}{3 \cdot A \cdot C - 4 \cdot B \cdot C}$$

печатать D

кон

Программа на языке Бейсик:

10 REM ВЫЧИСЛЕНИЕ ПО ФОРМУЛЕ

20 READ A, B, C

30 D=(4\*A-A\*B)/(3\*A\*C-4\*B\*C)

40 PRINT "РЕЗУЛЬТАТ D=";D

50 DATA 18, 183.5, 4.2

60 END

Результат вычисления:

РЕЗУЛЬТАТ D=1.131302

### ПРАКТИЧЕСКАЯ РАБОТА 5

**Тема:** Разветвляющийся алгоритм (оператор выбора). Запись разветвляющегося алгоритма на языке Бейсик, выполнение его на ПЭВМ.

**Цель работы:** Изучение оператора выбора, составление разветвляющихся алгоритмов и программ с использованием операторов условного и безусловного переходов.

**Продолжительность работы** - 2 ч.

Теоретические сведения. Оператор выбора и оператор ветвления в алгоритмическом языке записываются по-разному, а в языке Бейсик оба оператора имеют один и тот же аналог.

Оператор выбора на алгоритмическом языке записывается так:

выбор

при условии 1 : серия1

при условии 2 : серия2

...

при условии n : серияn

иначе серия

все

или в сокращенной форме:

выбор

при условии 1 : серия1

при условии 2 : серия2

...

при условии n : серияn

все

Оператор ветвления на языке Бейсик записывается так:

IF <условие> THEN <серия1> ELSE <серия2>

( IF - если, THEN - то, ELSE - иначе ).

Выполняя его, ПЭВМ проверяет соблюдение условия. Если оно соблюдено, выполняется серия1, если нет, то - серия2. Следует помнить, что серия1 в версии Бейсик-Агат может состоять только из одного оператора!

В языке Бейсик также возможна и сокращенная форма записи оператора выбора:

IF <условие> THEN <серия>

В этом случае при соблюдении условия выполнится один оператор, стоящий после THEN, а при несоблюдении - управление передается на следующую строку программы.

Отметим: для работы сокращенной формы оператора выбора вполне достаточно.

Рассмотрим, какие условия можно употреблять в операторе ветвления. Элементарные условия - это обычные отношения между выражениями:

<выражение1> <знак отношения> <выражение2>

В качестве знаков отношений используются знаки операций сравнения (табл. 8).

Т а б л и ц а 8

Знак операции сравнения	Значение знака	Знак операции сравнения	Значение знака
<	Меньше	<=	Меньше или равно
>	Больше	>=	Больше или равно
=	Равно	<>	Не равно

Три знака в третьей графе - составные, имеющие по два символа, так как на клавиатуре нет специальных клавиш  $\leq$ ,  $\geq$ ,  $\neq$ .

Оба выражения в условии должны быть одного типа: нельзя сравнивать число с символьной строкой!

Выполняя оператор выбора, ПЭВМ в первую очередь вычисляет значения обоих выражений, потом, сравнивая их, выясняет, соблюдено ли условие. Употребляя в выражениях числовые условия, особое внимание следует обратить на отношение "равно", так как может оказаться, что два числа, которые должны быть равны, все-таки отличаются (из-за ошибок округления).

Язык Бейсик позволяет применять операции сравнения не только к числовым, но и к символьным величинам. Сравнение их на совпадение (=) и несовпадение (<>) не требует комментариев. Если же символьные величины связаны знаками "больше" или "меньше", следует помнить, что коды латинских букв упорядочены по алфавиту - больший порядковый номер буквы в алфавите имеет больший код. Если длины строк не совпадают, то более короткая строка дополняется справа пробелом. Код символа "пробел" меньше кода любой буквы.

В версии языка Бейсик-Агат допускаются составные условия, образованные из элементарных условий, соединенных ключевыми словами: AND (и), OR (или) и NOT (не).

Приведенный ниже пример иллюстрирует соответствие записей на алгоритмическом языке (левый столбец) записям на языке Бейсик (правый столбец):

X=0 или X=1

(X=0) OR (X=1)

не (a=2 и b=3)

NOT (A=2 AND B=3)

Весь оператор выбора в языке Бейсик должен разместиться в одной строке программы. Однако не всегда удобно размещать в одну строку

серии операторов. В таких случаях используется оператор безусловного перехода GOTO (идти к), передающий управление на строку, с которой начинается нужный фрагмент программы:

GOTO < номер строки >

В результате его выполнения следующим выполняемым оператором будет оператор, номер которого указан за ключевым словом GOTO. Оператор безусловного перехода следует применять лишь в тех случаях, когда без него нельзя обойтись, так как слишком большое их число делает программу трудночитаемой.

#### Контрольные вопросы

1. В каких случаях применяется оператор выбора и как он выполняется?
2. Какие условия можно употреблять в операторе выбора?
3. Как образуются составные условия?
4. Как работает фрагмент программы:

```

10 PRINT "ВВЕДИТЕ НАТУРАЛЬНОЕ ЧИСЛО N"
20 INPUT N
30 IF (N > INT(N)) OR (N < 1) THEN 10
40 PRINT "СПАСИБО ,", N ; "ХОРОШЕЕ ЧИСЛО"

```

**Примечание.** Стандартная функция INT определяет целую часть числа N. Подробно применение стандартных функций будет рассмотрено в практической работе 7.

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задание 1 или 2 по вариантам;
- оформить отчет.

Отчет по практической работе 5 должен содержать:

- название и цель работы;
- запись общего вида оператора выбора;
- таблицу знаков отношений операций сравнения;
- запись общего вида оператора безусловного перехода;
- условие задания 1 или 2; блок-схему алгоритма; программы,

составленные на алгоритмическом языке и на языке Бейсик; результат работы программы.

**Задание 1.** Написать программу, реализующую следующий алгоритм:

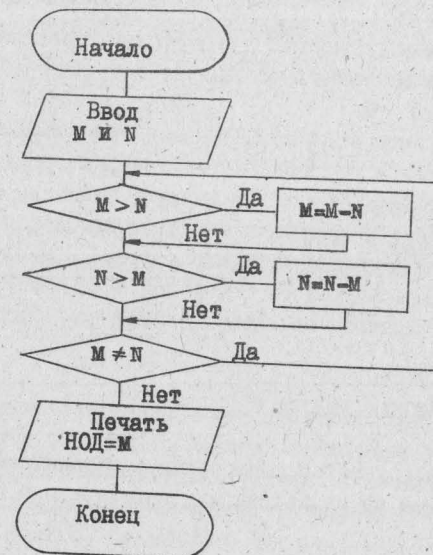
$$g = \begin{cases} e^t, & \text{если } 0 \leq t < 1; \\ e^{2t+0,1}, & \text{если } 1 \leq t < 2; \\ e^{t^2-1}, & \text{если } 2 \leq t < 3. \end{cases}$$

Значение t вводится пользователем, результат вывести на экран ВКУ.

**Задание 2.** Написать программу нахождения числа дней в месяце с учетом того, что известны (т.е. вводятся пользователем) номер месяца и признак високосного года  $A = \begin{cases} 1 & \text{для високосного года,} \\ \emptyset & \text{для невисокосного года.} \end{cases}$

**Пример выполнения задания.** Написать программу, реализующую алгоритм Евклида, основанный на том, что наибольший общий делитель (НОД) двух натуральных чисел совпадает с НОД пары, полученной из предыдущей заменой большего числа разностью большего и меньшего чисел.

#### Блок-схема





Программа на алгоритмическом языке:

```
алг НОД (нат m,n)
нач
  нат M,N
  M:=m
  N:=n
  пока M≠N
  нц
    выбор
      при M > N: M:=M-N
      при N > M: N:=N-M
    все
  кц
  знач:=M
кон
```

Программа на языке Бейсик:

```
10 REM *** НОД ***
20 PRINT "ВВЕДИТЕ НАТУРАЛЬНЫЕ ЧИСЛА M И N"
30 INPUT M,N
40 PRINT "M=";M,"N=";N
50 IF M > N THEN M=M-N
60 IF N > M THEN N=N-M
70 IF M<>N THEN GOTO 50
80 PRINT "НОД=";M
90 END
```

Результат работы программы:

```
ВВЕДИТЕ НАТУРАЛЬНЫЕ ЧИСЛА M И N
? 34,51
M=34          N=51
НОД -- 17
```

ПРАКТИЧЕСКАЯ РАБОТА 6

Тема: Циклический алгоритм (оператор повторения с параметром). Запись циклического алгоритма на языке Бейсик, выполнение его на ПЭЭМ.

Цель работы: Использование оператора повторения с параметром в циклических алгоритмах, отладка программ на ПЭЭМ.

Продолжительность работы - 2 ч.

Теоретические сведения. Оператор повторения имеет особое значение для построения сравнительно коротких алгоритмов, предписывающих многократное выполнение серии операторов.

Оператор повторения с параметром записывается следующим образом:

```
для x от x_min до x_max
нц
  серия
кц
```

где x - целочисленная переменная;

x<sub>min</sub> и x<sub>max</sub> - выражения, принимающие целочисленные значения.

В языке Бейсик оператор повторения записывается с помощью ключевых слов: FOR (для), TO (до), STEP (шаг), составляющих заголовок цикла, и ключевого слова NEXT (следующий), являющегося признаком конца цикла. Между заголовком и признаком конца располагается тело цикла.

Общий вид заголовка цикла:

FOR <имя переменной> = <выражение1> TO <выражение2> [STEP <выражение3>]

Часть заголовка, заключенная в квадратные скобки, необязательна. По умолчанию принимается стандартное значение шага, равное 1. Все выражения должны быть числовыми.

При выполнении цикла ПЭЭМ сначала отмечает для себя указанное имя счетчика цикла, присваивает в качестве начального значение выражения1, запоминает значение выражения2 (конечное значение счетчика). Если в заголовке цикла присутствует ключевое слово STEP, то запоминается значение выражения3 (шаг). Если STEP отсутствует, шаг получает значение, равное 1. Затем ПЭЭМ выполняет подряд следующие операторы, составляющие тело цикла до тех пор, пока не достигнет строки

NEXT <имя переменной>

в которой дано запомненное имя счетчика. При выполнении этого оператора к значению счетчика добавляется значение шага и проверяется, не превысило ли новое значение счетчика конечного значения. Если счетчик уже превысил конечное значение, то работа с циклом закончена, и ПЭЭМ выходит из цикла - "забывает" имя счетчика, конечное значение, шаг и приступает к выполнению оператора, следующего за оператором NEXT. Если

счетчик еще не достиг конечного значения, то ПЭВМ снова начинает выполнять тело цикла, начиная с оператора, стоящего за заголовком цикла - и все повторяется сначала. В любом случае тело цикла выполняется хотя бы один раз.

Тело цикла по отношению к остальной части программы замкнуто - не следует "входить" в него, минуя заголовок цикла, и "выходить" из него, минуя NEXT. Из цикла принято "выходить" только тогда, когда счетчик превысил конечное значение. Если надо выйти из цикла раньше, то следует присвоить счетчику значение, превышающее конечное значение, и оператором GOTO выйти на NEXT.

### Контрольные вопросы

1. В каких случаях используется оператор повторения?
2. Что такое счетчик цикла?
3. Что произойдет, если не указан шаг?
4. Каким будет результат работы программы:

```

10 S=0
20 FOR I=1 TO 10
30 S=S+I
40 PRINT "S=";S
50 END

```

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задание 1 или 2 по вариантам;
- оформить отчет.

Отчет по практической работе 6 должен содержать:

- название и цель работы;
- запись общего вида заголовка цикла и признака конца цикла;
- условие задания 1 или 2; блок-схему алгоритма; текст программы

на алгоритмическом языке и языке Бейсик; результат работы программы.  
**Задание 1.** Написать программу, которая составляет и печатает таблицу чисел значений функции

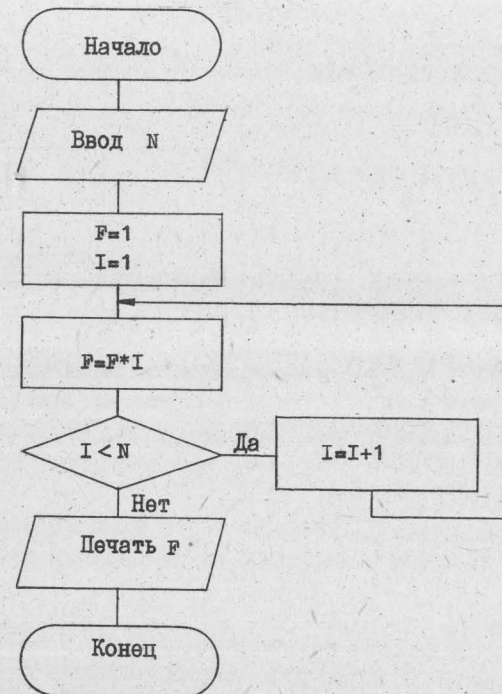
$$f(x) = \sum_{i=1}^m \sum_{j=1}^n (i+jx)^2$$

Значения  $m, n, l$  вводятся пользователем. Значение  $x$  меняется от 0 до 1.

**Задание 2.** Составить программу, которая суммирует не более 20 положительных чисел. Если введено отрицательное число или нуль, числа не суммируются. Количество суммируемых чисел задает пользователь (в пределах двадцати).

**Пример выполнения задания.** Составить программу вычисления факториала натурального числа, задаваемого пользователем.

### Блок-схема

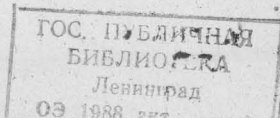


### Программа на алгоритмическом языке:

```

алг факториал (нат n)
нач цел i
  F:=1
  для i от 1 до n
  иц
    F:=F*i
  кц
кон

```



Программа на языке Бейсик:

```

10 REM ФАКТОРИАЛ
20 INPUT "ВВЕДИТЕ N";N
30 F=1
40 FOR I=1 TO N
50 F=F*I
60 NEXT I
70 PRINT "ФАКТОРИАЛ N;N;!=";F
80 END

```

Результат работы программы:

```

ВВЕДИТЕ N ? 6
ФАКТОРИАЛ 6!=720

```

ПРАКТИЧЕСКАЯ РАБОТА 7

**Тема:** Вспомогательные алгоритмы вычисления значений функций, выполнение их на ПЭВМ.

**Цель работы:** Ознакомление со стандартными вспомогательными функциями языка Бейсик и функциями, определяемыми программистом.

**Продолжительность работы** - 2 ч.

**Теоретические сведения.** Наиболее часто используемыми вспомогательными алгоритмами являются стандартные математические функции (табл. 9).

Т а б л и ц а 9

Имя функции	Выполняемое действие
SIN(X)	Вычисляет синус аргумента $X^*$
COS(X)	Вычисляет косинус аргумента X

\* Значение аргумента X в тригонометрических функциях задается в радианах.

Имя функции	Выполняемое действие
TAN(X)	Вычисляет тангенс аргумента X
ATN(X)	Вычисляет арктангенс аргумента X
SQR(X)	Вычисляет положительный квадратный корень аргумента $X(X \geq 0)$
INT(X)	Определяет целую часть аргумента X
ABS(X)	Определяет абсолютное значение аргумента X
EXP(X)	Вычисляет показательную функцию $e^x$ ( $e = 2,71228, x \leq 37$ )
LOG(X)	Вычисляет натуральный логарифм аргумента X
SGN(X)	Принимает значения: +1 при $X > 0$ -1 при $X < 0$ $0$ при $X = 0$
RND(X)	Принимает псевдослучайное значение в интервале от 0 до 1. При $X > 0$ - элемент случайной последовательности, при $X < 0$ - запуск псевдослучайной последовательности (для каждого X), при $X=0$ - повторение последнего результата

Для обращения к стандартным функциям необходимо набрать имя функции и аргумент, заключенный в скобки. Функции используются в выражениях так же, как константы и переменные.

Функция вычисляет результат и возвращает его значение. Интерпретатор продолжает вычислять значение выражения, как если бы вместо функции указали результат ее вычисления.

Пользователь для преобразования радиан в градусы может использовать следующую формулу:

$$\text{значения в радианах} = \frac{\text{значения в градусах} \cdot 3,1416}{180}$$

Кроме встроенных стандартных математических функций в качестве вспомогательных алгоритмов выступают функции, определяемые программистом. Если в программе необходимо многократно вычислять значение функции, которая не является стандартной функцией языка Бейсик, то программист может сам определить ее и затем использовать в своей программе так же, как и стандартные.

Имя функции программиста должно состоять из трех букв латинского алфавита, первые две из них — FN. Таким образом можно определить 26 числовых функций с именами FNA, ..., FNZ и 26 литерных функций с именами FNAH, ..., FNZH.

Функцию можно определить оператором DEFINE (определить):

DEF < имя функции > ( < аргумент > ) = < выражение >

В ходе выполнения программы операторы DEF пропускаются, однако каждый раз, когда необходимо значение функции программиста, это значение будет вычислено согласно определению функции в соответствующем операторе DEF.

Аргументом является формальный параметр функции, а выражение представляет собой записанный в виде формулы алгоритм ее вычисления. При использовании функции вместо формального параметра употребляется фактический. Именно его конкретное значение берется при вычислении значения функции. Переменные, имена которых встречаются в определяющем функцию выражении, но не упомянуты в списке аргументов, при вычислении значения функции заменяются текущими значениями соответствующих переменных.

Оператор DEF относится к числу невыполняемых. Поэтому он может появиться в любом месте программы, но необходимо только контролировать, чтобы он предшествовал первому обращению к функции.

#### Контрольные вопросы

1. Какие вспомогательные алгоритмы используются в языке Бейсик?
2. Как обратиться к функции в программе?
3. В каких случаях программист заинтересован в определении новой функции?

4. Каков будет результат работы программы:

10 A=10

20 DEF FNA(X)=X\*X-A

30 DEF FNB(Y)=5+Y

40 PRINT FNB(FNA(4))

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задание 1 или 2 по вариантам;
- оформить отчет.

Отчет по практической работе 7 должен содержать:

- название и цель работы;
- таблицу встроенных стандартных математических функций;
- запись общего вида оператора описания функции программиста;
- условие задания 1 или 2, составленную блок-схему алгоритма, программы на алгоритмическом языке и на языке Бейсик.

Задание 1. Составить программу вычисления значения функции

$$f(x) = \ln(x + \sqrt{x^2 + 1}) \quad \text{при } x = -1, 8, 15, 20.$$

Вычисление  $f(x)$  оформить как функцию.

Задание 2. Составить программу вычисления значения функции

$$f(y) = \frac{1}{\sqrt{2y}} - \frac{y^2}{2} f(y) \quad \text{при } y = 0.8; 3; 5.6.$$

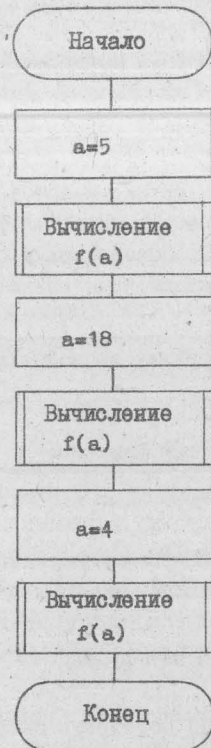
Вычисление  $f(y)$  оформить как функцию.

Пример выполнения задания. Составить программу вычисления значений функции

$$f(a) = \operatorname{tg}(a) - \frac{\sqrt{a^2 - 3}}{2} \quad \text{при } a = 5; 18; 4.$$

Вычисление  $f(a)$  оформить как функцию.

Блок-схема



Программа на алгоритмическом языке:

алг вычисление (вещ A, F)

вг A

рез F

нач A:=0.5

FUN(A, F)

печатать F

A:=18

FUN(A, F)

печатать F

A:=4

FUN(A, F)

печатать F

кон

алг FUN (вещ X, Y)

вг X

рез Y

нач

Y:=-tg(X)- $\frac{\sqrt{X^2-3}}{2}$

кон

Программа на языке Бейсик:

10 REM ВЫЧИСЛЕНИЕ ФУНКЦИИ

20 DATA 5,18,4

25 DEF FNF(A)=TAN(A)-SQR(A^2-3)/2

30 FOR I=1 TO 3

40 READ A

50 R=FNF(A)

60 PRINT R

70 NEXT I

80 END

Результат работы программы:

-5.72572289

-10.0955502

-.644954357

ПРАКТИЧЕСКАЯ РАБОТА 8

Тема: Работа с литерными величинами. Функции обработки литерных величин в языке Бейсик.

Цель работы: Ознакомление с методами обработки литерных величин с применением специальных функций обработки строк.

Продолжительность работы - 2 ч.

Теоретические сведения. В отличие от числовых выражений символьные более просты в использовании, поскольку имеют всего лишь одну операцию - сложение, или конкатенацию. Результат вычисления символьного выражения - всегда строка символов.

Операции над строками не ограничиваются их сложением. Возможности преобразования строк значительно расширяются с помощью так называемых строковых функций, приведенных в табл. 10.

Т а б л и ц а 10

Название функции	Действие, выполняемое функцией
LEN(A <sub>x</sub> )	Подсчет числа символов в строке, указанной в скобках
MID(A <sub>x</sub> , X, Y)	Выделение подстроки в строке, указанной в скобках в качестве первого аргумента (X - номер символа строки, с которого начинается выделение; Y - количество выделяемых символов)
VAL(A <sub>x</sub> )	Преобразование строковой величины в числовую, равную записанному в строке A <sub>x</sub> числу
STR(X)	Преобразование числа (целого или вещественного), соответствующего переменной X, в строковую переменную, т.е. в запись этого числа в виде соответствующей цепочки цифр и, если необходимо, других употребляемых символов (десятичной точки, знаков "плюс" и "минус", символа E)
LEFT(A <sub>x</sub> , X)	Выделение подстроки в строке, указанной в качестве первого аргумента. Вторым аргументом указывает количество выделяемых символов, начиная с начала строки, т.е. X первых символов строки A <sub>x</sub>
RIGHT(A <sub>x</sub> , X)	Выделение подстроки в строке, указанной в качестве первого аргумента. Вторым аргументом указывает количество выделяемых символов в конце строки, т.е. X последних символов строки A <sub>x</sub>

Следует заметить, что если название функции завершается знаком  $\times$  (солнышко), то значением этой функции является строка символов. Значением функций, не имеющих этого знака в конце названия, является число.

Функции LEFT $\times$  и RIGHT $\times$  отсекают соответственно от левого и правого концов строковой переменной такое количество знаков, которое указано в скобках в качестве второго аргумента, например:

40 A $\times$  = "ПАССАЖИР"

50 B $\times$  = LEFT $\times$ (A $\times$ , 3)

60 C $\times$  = RIGHT $\times$ (A $\times$ , 3)

70 PRINT B $\times$ , C $\times$

После выполнения такого фрагмента на экране ВКУ появится:

П А С                      Ж И Р

Функция LEN довольно часто применяется в работе с текстовыми величинами. Результат ее вычисления - число.

Рассмотрим фрагмент программы:

30 S = 0

40 A $\times$  = "КРОКОДИЛ ГЕНА"

50 PRINT A $\times$

60 R = LEN(A $\times$ )

70 PRINT R; "\_СИМВОЛОВ"

80 FOR I = 1 TO R

90 IF MID $\times$ (A $\times$ , I, 1) < > "\_" THEN 110

100 S = S + 1

110 NEXT I

120 PRINT "ТЕКСТ СОДЕРЖИТ -"; S; "\_ ПРОБЕЛОВ"

После выполнения этой программы на экране ВКУ появится:

КРОКОДИЛ ГЕНА

13 СИМВОЛОВ

ТЕКСТ СОДЕРЖИТ - 1 ПРОБЕЛ

На примере этого фрагмента видно, что функция LEN подсчитывает все символы, включая пробелы. Функция MID $\times$  результатом своего выполнения имеет один символ (задается третьим аргументом в скобках) и применяется в выражениях так же, как и переменные.

Функции VAL и STR $\times$  служат для преобразования данных одного вида в данные другого вида.

Пусть в строке A $\times$  записано некоторое число, например A $\times$  = "1987".

Функция VAL(A<sub>ж</sub>) преобразует эту строковую переменную в числовую - VAL(A<sub>ж</sub>)=1987.

Пусть теперь, наоборот, у нас есть число A=83E2. Функция STR<sub>ж</sub>(A) преобразует его в строковую переменную, в запись этого числа в виде соответствующей цепочки цифр и, если требуется, десятичной точки, знаков и символа E. В нашем примере STR<sub>ж</sub>(A)="83E2".

#### Контрольные вопросы

1. Какие операции допустимы над строковыми величинами?
2. Что означает знак ж в конце имени функции?
3. Какие строковые функции Вы знаете?
4. Каким будет результат работы программы:
  - 1ж A1ж="ЖЕ "
  - 2ж A2ж="РА "
  - 3ж A3ж="ТА "
  - 4ж Bж=A1ж+A3ж
  - 5ж Cж=A3ж+A2ж
  - 6ж Dж=A2ж+A1ж+A3ж
  - 7ж PRINT Bж,Cж,Dж
  - 8ж END

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задание 1 или 2 по вариантам;
- оформить отчет.

Отчет по практической работе 8 должен содержать:

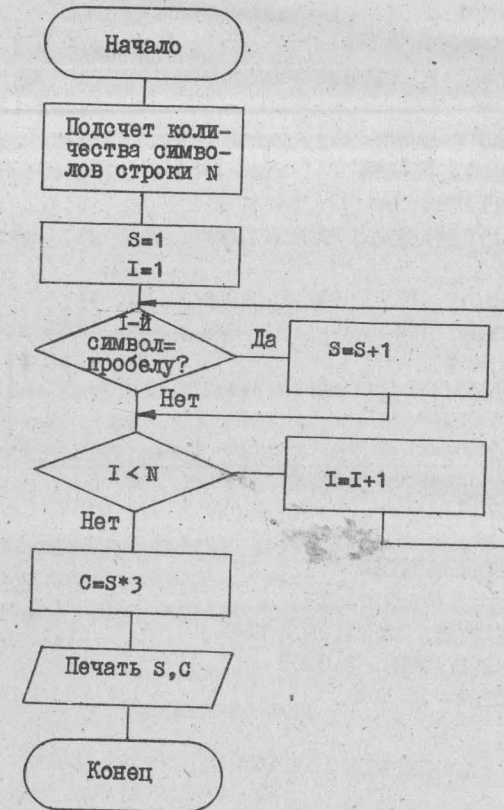
- название и цель работы;
- таблицу строковых функций;
- условие задания 1 или 2, блок-схему алгоритма, программу на алгоритмическом языке и на языке Бейсик.

**Задание 1.** Проверить, изменятся ли слова КАЗАК и ПОТОП, прочитанные слева направо. Составить программу, используя функции LEFT и RIGHT.

**Задание 2.** Подсчитать количество гласных букв в слове "прозаседавшиеся". Составить программу, используя функцию MID.

**Пример выполнения задания.** Подсчитать количество слов в телеграмме и ее стоимость. Стоимость одного слова - 3 коп.

#### Блок-схема



#### Программа на алгоритмическом языке:

алг ТЕЛЕГРАММА (лит X, цел S, C)  
нач

```

    цел i
    S:=1
    для i от 1 до длин (X)
    ни
  
```

```

ЕСЛИ X [i:i] = "_"
  ТО S:=S+1
ВСЕ
КЦ
C:=S*3

```

КОН

Программа на языке Бейсик:

```

10 REM ТЕЛЕГРАММА
20 PRINT "ВВЕДИТЕ СТРОКУ ТЕЛЕГРАММЫ"
30 INPUT X$
40 N=LEN(X$):S=1
45 IF N=0 THEN 110
50 FOR I=1 TO N
60 IF MID$(X$,I,1)="_" THEN S=S+1
70 NEXT I
80 C=S*3
90 PRINT "В ВАШЕЙ ТЕЛЕГРАММЕ - ";S;" СЛОВ"
100 PRINT "ЕЕ СТОИМОСТЬ -";C;" КОП".

```

Результат работы программ:

```

ВВЕДИТЕ СТРОКУ ТЕЛЕГРАММЫ
?ВСТРЕЧАЙТЕ ПОЕЗД 58 ВАГОН 6 МАМА
В ВАШЕЙ ТЕЛЕГРАММЕ - 6 СЛОВ
ЕЕ СТОИМОСТЬ - 18 КОП

```

### ПРАКТИЧЕСКАЯ РАБОТА 9

Тема: Обработка табличной информации на ПЭЕМ.

Цель работ: Изучение команды описания массивов, переменных с индексами и применение их при решении задач.

Продолжительность работ - 2 ч.

Теоретические сведения. Кроме простых переменных в языке Бейсик и в алгоритмическом языке используются переменные с индексами, являющиеся элементами массивов (или табличными величинами).

Линейные таблицы - это одномерные массивы (векторы). Им соответствуют переменные с одним индексом.

Прямоугольные таблицы - это двумерные массивы (матрицы). Им соответствуют переменные с двумя индексами: первый индекс - номер строки, второй - номер столбца.

Индексом может быть целая константа, или переменная, или любое выражение, которое в результате вычисления дает целое значение.

Индексная переменная записывается в виде имени, за которым следует одна или две индексные величины, заключенные в круглые скобки. Если индексов два, то они разделяются запятой, например: A(8,6); B(M,K); B(T+P); A1(8).

Все массивы должны быть описаны, т.е. указана размерность массива с помощью оператора DIM (размер):

DIM <имя> (разм)

В одном операторе DIM могут быть указаны несколько массивов через запятую, например:

10 DIM A(30,10),B(10),C(2,2)

Следует помнить, что переменные, объединенные в массив, имеют один и тот же тип.

Массив может быть и многомерным. Но в любом случае количество индексов в переменной должно соответствовать размерности массива.

Массивы могут быть как числовыми, так и символьными. Во втором случае после имени массива ставится знак \$, например: A\$(3), ВС\$(1,2), Т\$(1,1,2).

Наименьшее значение индекса равно нулю, а наибольшее определяется размером массива минус единица. Например, массив описан как DIM A(5). Элементы массива A будут иметь индексы от 0 до 4, а именно: A(0), A(1), A(2), A(3), A(4).

### Контрольные вопросы

1. Какие типы массивов Вы знаете?
2. Что может являться индексом?
3. Каким оператором описывается массив?
4. Каким будет результат работы программы:

```

10 REM МАССИВ
20 DATA 8,5, 3.8, 16, 18.2, 35
30 DIM C(6): S=0
40 FOR I=1 TO 6
50 READ C(1): S=S+C(1)
60 NEXT I
70 PRINT S
80 END

```



Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задание 1 или 2 по вариантам;
- оформить отчет.

Отчет по практической работе 9 должен содержать:

- название и цель работы;
- ответ на вопрос 4;
- запись общего вида оператора описания массива;
- условие задания 1 или 2, блок-схему алгоритма, программы на алгоритмическом языке и на языке Бейсик, результат выполнения программы.

**Задание 1.** Имеется каталог домашней библиотеки, задаваемый оператором DATA. Каждая запись в каталоге состоит из 30 позиций: 12 позиций занимают сведения об авторе книги, 15 позиций - о названии, 3 позиции - о количестве томов. Подсчитать, сколько всего томов содержит библиотека.

**Задание 2.** Определить наивысшую среднесуточную температуру за месяц (в месяце 30 дней), если замеры температуры делались каждые три часа. Значения температур занесены в сводную месячную таблицу, состоящую из 30 строк и 8 столбцов.

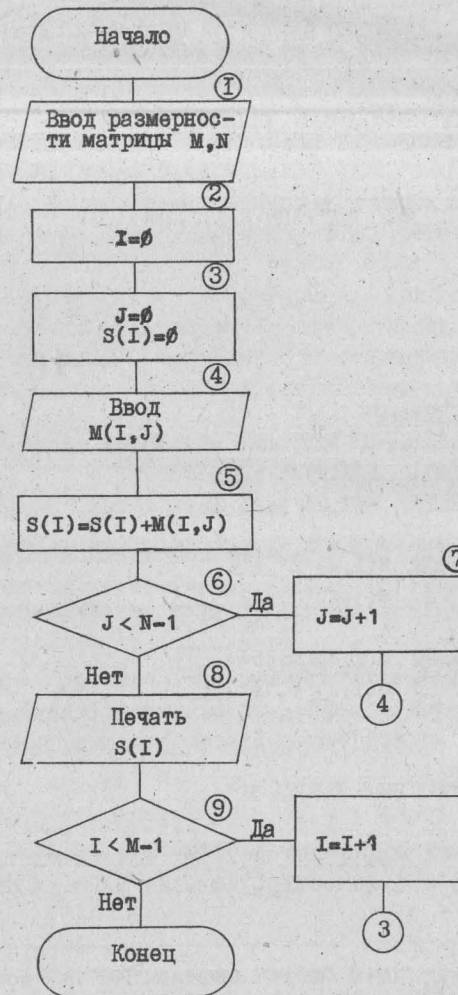
**Пример выполнения задания.** Составить программу нахождения суммы элементов матрицы по строкам.

Программа на алгоритмическом языке:

```

алг сумма (вещ таб М [1:m, 1:n],  вещ таб S [1:m])
  вар М
  рез S
нач  цел i, j
    i:=1
    пока i ≤ m
      ни
        пока j ≤ n
          ни
            S[i] := S[i] + M[i, j]
            j := j + 1
          кн
        i := i + 1
      кн
    кн
кон
  
```

Блок-схема



Программа на языке Бейсик:

```

10 REM СУММА ЭЛЕМЕНТОВ ПО СТРОКАМ
20 INPUT "ВВЕДИТЕ КОЛИЧЕСТВО ЭЛЕМЕНТОВ МАТРИЦЫ М И N:";
M,N
30 DIM M(M,N),S(M)
40 FOR I=0 TO M-1
50 S(I)=0
60 FOR J=0 TO M-1
70 INPUT "ВВЕДИТЕ ЭЛЕМЕНТ МАТРИЦЫ M(I,J);M(I,J)
80 S(I)=S(I)+M(I,J)
90 NEXT J
100 PRINT S(I)
110 NEXT I
120 END
    
```

Результат работы программы:

```

ВВЕДИТЕ КОЛИЧЕСТВО ЭЛЕМЕНТОВ МАТРИЦЫ М И N:3,2 
ВВЕДИТЕ ЭЛЕМЕНТ МАТРИЦЫ M(I,J) 5 
ВВЕДИТЕ ЭЛЕМЕНТ МАТРИЦЫ M(I,J) 3.2 
8.2
ВВЕДИТЕ ЭЛЕМЕНТ МАТРИЦЫ M(I,J) 2 
ВВЕДИТЕ ЭЛЕМЕНТ МАТРИЦЫ M(I,J) 13.62 
15.62
ВВЕДИТЕ ЭЛЕМЕНТ МАТРИЦЫ M(I,J) 0.654 
ВВЕДИТЕ ЭЛЕМЕНТ МАТРИЦЫ M(I,J) 0.101 
0.755
    
```

**ПРАКТИЧЕСКАЯ РАБОТА 10**

Тема: Обработка графической информации на ПЭМ.

Цель работы: Ознакомление с графическими режимами работы и операторами графики.

Продолжительность работы - 2 ч.

Теоретические сведения. ПЭМ "Агат" оснащена ВКУ цветного отображения, поэтому решение графических задач может быть наглядным.

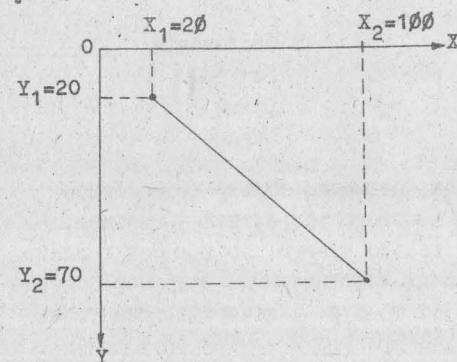
В отличие от алгоритмического языка, в котором существуют операторы "Вперед", "Назад", "Вправо", "Влево", "Рисуй", "Не рисуй", язык Бейсик - Агат имеет лишь один оператор PLOT (график), заменяющий все вышеперечисленные:

PLOT <координата1> [TO <координата2>]

Часть, заключенная в квадратные скобки, необязательна. Если в операторе PLOT указана только координата1, то на экране ВКУ будет выведена точка. Место расположения этой точки определяется значением координаты1.

Под координатой понимается пара значений, соответствующих положению точки относительно оси X и оси Y, перечисленных через запятую, например: PLOT 100,20.

В отличие от системы координат, применяемой в математике, в языке Бейсик система координат имеет точку отсчета из верхнего левого угла.



Если оператор PLOT не содержит часть, заключенную в квадратные скобки, то будет начерчена прямая линия от точки с <координатой1> до точки с <координатой2>, например:  
PLOT 20,20 TO 100,70

Следует отметить, что работа программы на языке Бейсик с графикой начинается с включения графического режима. В языке Бейсик-Агат предусмотрено три графических режима (табл. II).

Таблица II

Обозначение графического режима	Назначение графического режима	Диапазон допустимых значений координат	Диапазон номеров страниц
GR	Включение графики низкого разрешения (64 x 64)	0 + 63	2 + 31
MGR	Включение графики среднего разрешения (128 x 128)	0 + 127	1 + 7
HGR	Включение графики высокого разрешения (256 x 256)	0 + 255	1 + 7

Оператор включения графического режима может иметь одну из трех форм:

GR=<N>, или MGR=<E>, или HGR=<N> ,

где N - номер страницы графического режима.

Чтобы отображение графической информации появилось на экране ВКУ, задается цвет точки или линии, которую мы выводим. Для этого используется оператор COLOR (цвет):

COLOR = <номер цвета >

Соответствие номеров цвету приведено ниже.

Номер	Цвет	Номер	Цвет
1	Красный	5	Фиолетовый
2	Зеленый	6	Голубой
3	Желтый	7	Белый
4	Синий	8	Черный

В графике низкого разрешения (GR) возможно только черно-белое отображение, а в графике среднего и высокого разрешения - цветное.

Оператор PLOT может чертить не только прямую, но и ломаную линию. Для этого задается несколько точек с их координатами, например оператор 12 PLOT 0, 0 TO 127, 127 TO 0, 127 TO 0, 0 начертит прямоугольный треугольник.

Ограничением в продолжении оператора PLOT служат 255 символов, которые может содержать одна строка программы.

Если координаты начальной точки не заданы, то используется конечная точка последнего по выполнению оператора PLOT.

Переход из графического режима в текстовой осуществляется оператором TEXT (текст):

TEXT = <номер страницы >

Оператор TEXT устанавливает текстовый режим. Диапазон изменения номеров страниц 2 - 31.

Соединение текстовой и графической страницы (рисунок и текст) недопустимо.

Включение графического режима всегда очищает экран, однако при переходе к текстовому режиму экран не очищается, а новая информация накладывается на предыдущую.

При использовании двух режимов работы (текстового и графического) возникает вопрос: как задержать информацию на экране? Ведь если не

будет предусмотрена задержка каждой страницы выводимой информации, то она промелькнет очень быстро, и мы увидим лишь последнюю страницу. Для этой цели служит оператор GET (извлечь):

GET <имя литерной переменной >

Доходя до этого оператора, выполнение программы приостанавливается, и ПЭМ ожидает сигнала с клавиатуры. Для подачи этого сигнала нужно нажать любую информационную клавишу. Введенный символ на экране не отображается. После ввода символа выполнение программы будет продолжено.

#### Контрольные вопросы

1. В какой последовательности должны располагаться операторы графического режима?

2. Как начертить ломаную линию?

3. Что будет выведено на экран ВКУ в результате работы программы:

1) MGR=3

2) PLOT 100, 20 TO 100, 80

3) COLOR=2

4) PLOT 100, 80 TO 20, 80

4. Каково назначение оператора GET?

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;

- ответить на контрольные вопросы;

- выполнить задание 1 или 2 по вариантам;

- оформить отчет.

Отчет по практической работе 10 должен содержать:

- название и цель работы;

- ответ на вопрос 3;

- таблицу графических режимов и таблицу цветов;

- запись общего вида операторов включения графического и текстового режимов, COLOR, PLOT и GET;

- условие задания 1 или 2, программы на алгоритмическом языке

и на языке Бейсик.

Задание 1. Написать программу вывода в режиме MGR пятиконечной звезды красного цвета.

Задание 2. Написать программу вывода в режиме HGR домика на курьих ножках, используя различные цвета.

Пример выполнения задания. Написать программу вывода в режиме HGR треугольника, заштрихованного белым цветом.

Программа на алгоритмическом языке:

алг треугольник (вещ N)

арг N

нач

цел i

i:=0

не рисуй

вперед (10); направо (90); вперед (10)

налево (90)

C=10

пока i ≤ (N - 10) / 2

нц

i:=1

рисуй

вперед (N - C); направо (90)

не рисуй

вперед (1); направо (90), назад (1)

рисуй

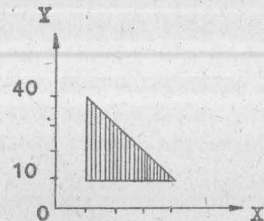
назад (N - C - 1); налево (90); вперед (1); налево (90)

i = i + 1

C = C + 1

кц

кон



Программа на языке Бейсик:

10 REM ТРЕУГОЛЬНИК

20 INPUT :REM ЗНАЧЕНИЕ КРАЙНЕЙ ТОЧКИ

30 IF N > 255 THEN 100

40 HGR=3

50 COLOR=7

60 FOR I=10 TO N

70 PLOT 1,1 TO 1,N

80 NEXT I

90 GOTO 100

100 TEXT=3

110 PRINT "КООРДИНАТА ПРЕВЫШАЕТ МАКСИМАЛЬНОЕ ЗНАЧЕНИЕ"

120 END

Результат работы программы:

RUN

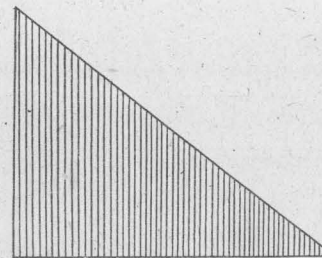
?266

КООРДИНАТА ПРЕВЫШАЕТ МАКСИМАЛЬНОЕ

ЗНАЧЕНИЕ

RUN

?234



#### ПРАКТИЧЕСКАЯ РАБОТА 11

Тема: Запись вспомогательных алгоритмов в виде подпрограммы, выполнение их на ПЭВМ.

Цель работы: Ознакомление с записью вспомогательных алгоритмов в виде подпрограмм.

Продолжительность работы - 2 ч.

Теоретические сведения. В практической работе 7 Вы научились записывать вспомогательные алгоритмы в виде функций, использовать стандартные математические функции. Но могут возникнуть ситуации, когда нужную для расчета функцию не удастся выразить с помощью одних лишь арифметических операций и встроенных функций и для ее вычисления одного оператора присваивания недостаточно.

В таких случаях вспомогательный алгоритм оформляют в виде подпрограммы. Использование подпрограмм, как и любого вспомогательного алгоритма, целесообразно при составлении сложных программ, когда определенный фрагмент программы используется неоднократно.

Переход на подпрограмму осуществляется по оператору GO to SUBroutine (идти к подпрограмме):

GOSUB <номер строки>

Выполняя этот оператор, ПЭВМ запоминает "адрес возврата" - номер строки, следующий за данным оператором GOSUB, и приступает к выполне-

или операторов, записанных со строки с указанным в GOSUB номером. Только встретив оператор RETURN (возвращаться), ПЭЕМ возвращается к запомненному адресу. Таким образом, оператор RETURN служит признаком конца подпрограммы.

Одна подпрограмма может вызвать другую подпрограмму, другая - третью и т.д.

Передавать значения параметров подпрограммы главной программе должен сам программист.

#### Контрольные вопросы

1. В каких случаях применяются подпрограммы?
2. Чем отличаются операторы GOSUB и GOTO?
3. Рассказать о всех возможных вспомогательных алгоритмах, применяемых в языке Бейсик.

4. Каким будет результат работы программы:

```

100 M=186:W=0.5
110 GOSUB 200
120 PRINT "СКОРОСТЬ САМОЛЕТА="; V
130 M=350:W=3.5
140 GOSUB 200
150 PRINT "СКОРОСТЬ ПЕЗДА="; V
160 END
200 V=M/W
210 RETURN
    
```

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задание 1 или 2 по вариантам;
- оформить отчет.

Отчет по практической работе II должен содержать:

- название и цель работы;
- запись общего вида оператора перехода на подпрограмму и оператора возврата в основную программу;
- условие задания 1 или 2, блок-схему алгоритма, программы на алгоритмическом языке и на языке Бейсик, результат работы программы.

Задание 1. Рассчитать формулу

$$y = \frac{\cos(1) \cdot \sin(b)}{\sin(21+b)}$$

если значения 1 и b заданы в градусах. Для перевода угла, заданного в градусах, в радианы использовать подпрограмму.

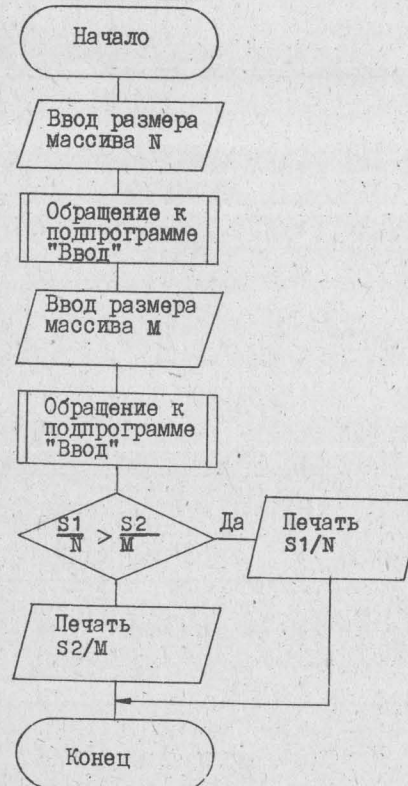
Задание 2. Рассчитать формулу

$$y = \frac{\sum_{i=1}^6 a_i \cdot 8\pi}{\sum_{j=1}^5 b_j \cdot 8\pi} \cdot 180,$$

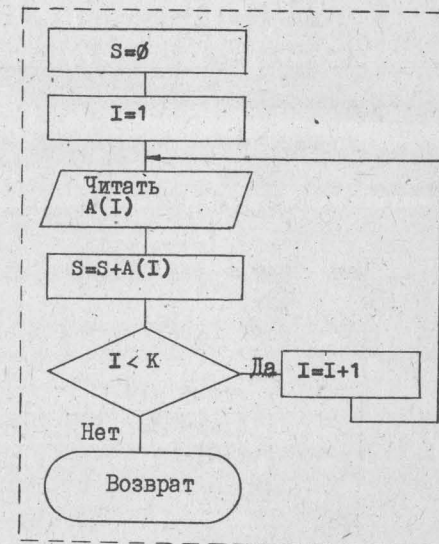
если значения  $a_i$  и  $b_j$  - элементы одномерных массивов. Вычисление произведения оформить подпрограммой.

Пример выполнения задания. Ввести два одномерных массива чисел. Определить большее среднее арифметическое элементов массивов.

#### Блок-схема



#### Подпрограмма "Ввод"



Программа на алгоритмическом языке:

алг сравнение (вещ таб A[1:n], B[1:m], вещ P)

арг A, B

рез P

нач цел N, M вещ S1, S2, цел K

K:=N

ВВОД (K, S)

S1:=S

K:=M

ВВОД (K, S)

S2:=S

если  $\frac{S1}{N} > \frac{S2}{M}$

то P:=  $\frac{S1}{N}$

иначе P:=  $\frac{S2}{M}$

кон

алг ВВОД (цел K, вещ S)

арг K

рез S

нач цел i вещ таб A[1:K]

i:=1

пока i ≤ K

нц

читать A[i]

S:=S+A[i]

i:=i+1

кц

кон

Программа на языке Бейсик:

10 REM СРАВНЕНИЕ СРЕДНИХ АРИФМЕТИЧЕСКИХ

20 INPUT "ВВЕСТИ РАЗМЕРНОСТЬ 1-ГО МАССИВА"; K

40 GOSUB 200

50 S1=S/K

60 INPUT "ВВЕСТИ РАЗМЕРНОСТЬ 2-ГО МАССИВА"; K

70 GOSUB 200

80 S2=S/K

90 IF S1 > S2 THEN 120

100 PRINT "БОЛЬШИМ ЯВЛЯЕТСЯ СРЕДНЕЕ АРИФМЕТИЧЕСКОЕ ВТОРОГО МАССИВА-"; S2

110 GOTO 130

120 PRINT "БОЛЬШИМ ЯВЛЯЕТСЯ СРЕДНЕЕ АРИФМЕТИЧЕСКОЕ ПЕРВОГО МАССИВА-"; S1

130 END

200 S=0

210 FOR I=1 TO K

220 INPUT A(I)

230 S=S+A(I)

240 NEXT I

250 RETURN

Результат работы программы:

ВВЕСТИ РАЗМЕРНОСТЬ 1-ГО МАССИВА 3

? 5

? 8

? 3

ВВЕСТИ РАЗМЕРНОСТЬ 2-ГО МАССИВА 2

? 18

? 30

БОЛЬШИМ ЯВЛЯЕТСЯ СРЕДНЕЕ АРИФМЕТИЧЕСКОЕ ВТОРОГО МАССИВА - 24.

ПРАКТИЧЕСКАЯ РАБОТА 12

Тема: Режимы вывода текстов на экран ВКУ ПЭЕМ.

Цель работы: Ознакомление с основными режимами вывода текстов на экран ВКУ, скоростью вывода текстов и возможностями цветового вывода.

Продолжительность работы - 2 ч.

Теоретические сведения. В версии Бейсик-Агат существуют три режима вывода текстов на экран ВКУ: мерцающий, инверсный и прямой.

Для включения мерцающего режима используется оператор FLASH (мелькать). Все стоящие за оператором включения режима операторы PRINT будут выводить текст в мерцающем режиме до тех пор, пока режим вывода не будет изменен.

Инверсный режим вывода текста включается оператором INVERSE (обратный). Режим обеспечивает вывод текста темным шрифтом по светлomu фону.

Прямой режим вывода текста (светлый по черному фону) включается оператором NORMAL (нормальный). Если режим вывода текста не задан, то по умолчанию он принимается прямым - светлым по черному фону.

Существует возможность задания скорости вывода текста с помощью оператора SPEED (скорость):

SPEED = <скорость>

Диапазон изменения скорости вывода 0 - 255.

Можно использовать цвет для вывода текста. Для этого применяется оператор RIBBON (лента):

RIBBON = <номер цвета>

Номера цветов см. в таблице, приведенной в практической работе 10.

Если цвет не задан, то вывод текста осуществляется светлым по черному. Если задан оператор RIBBON, то текст будет выводиться заданным цветом до тех пор, пока не будет выдан новый оператор RIBBON, меняющий цвет. Если после оператора RIBBON задан оператор INVERSE, то вывод текста будет осуществляться по фону того цвета, который задан оператором RIBBON.

#### Контрольные вопросы

1. Какие режимы вывода текстов Вы знаете?
2. Как изменить скорость вывода текста на экран ВКУ?
3. Как задать вывод текста по зеленому фону?
4. Как будет выведен текст на экран ВКУ:

100 RIBBON=5

110 INVERSE

120 PRINT "РАБОТА ОКОНЧЕНА !"

130 END

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задание 1 или 2 по вариантам;
- оформить отчет.

Отчет по практической работе 12 должен содержать:

- название и цель работы;
- запись общего вида операторов задания режимов вывода текста, оператора задания скорости вывода и цвета;
- условие задания 1 или 2, текст программы на языке Бейсик.

Задание 1. Выдать на экран ВКУ свои анкетные данные (ФИО, год рождения, месяц, число, номер школы, класс) со скоростью 200 по голубому фону.

Задание 2. Выдать на экран ВКУ формулировку теоремы Пифагора со скоростью 60 синим шрифтом.

Пример выполнения задания. Выдать на экран ВКУ четверостишие А.С.Пушкина со скоростью 85. Фон - зеленый.

Программа на языке Бейсик:

```
10 REM ТЕКСТ
20 RIBBON=2
30 INVERSE
40 PRINT:PRINT "КАВКАЗ ПОДО МНОЮ. ОДИН В ВЫШИНЕ"
50 PRINT:PRINT "СТОЮ НАД СНЕГАМИ У КРАЯ"
60 PRINT:PRINT "                СТРЕМНИНЫ;"
70 PRINT:PRINT "ОРЕЛ, С ОТДАЛЕННОЙ ПОДНЯВШИСЬ"
80 PRINT:PRINT "                ВЕРШИНЫ;"
90 PRINT:PRINT "ПАРИТ НЕПОДВИЖНО СО МНОЙ НАРАВНЕ"
100 PRINT:PRINT "                А.С.ПУШКИН"
110 END
```

#### ПРАКТИЧЕСКАЯ РАБОТА 13

Тема: Операторы размещения информации на экране ВКУ.

Цель работы: Ознакомление с дополнительными возможностями оператора PRINT, операторами очистки экрана и установления курсора в нужную позицию.

Продолжительность работы - 2 ч.

Теоретические сведения. В практической работе 4 было рассмотрено размещение информации на экране ВКУ оператором PRINT с помощью разделителей "запятая" и "точка с запятой". Варьирование разделителями дает довольно большие возможности для компоновки выводимых строк. Еще более расширяет эти возможности использование специальных функций, являющихся элементами оператора PRINT.

Функция TAB(X) предписывает процессору начать вывод очередного значения, начиная с X-й позиции строки. X – целое положительное число. Оно должно быть меньше максимального количества позиций в строке. Применяя эту функцию, можно двигаться только вправо. Поэтому, если позиция, определяемая функцией TAB, меньше текущей позиции строки, то функция игнорируется. Например, фрагмент программы

```
1Ø A=1Ø.5
2Ø B=1.612
3Ø C=362.14
4Ø PRINT A, TAB(7),B; TAB(21);C
```

выводит на экран:

```
10.5      1.612      362.14
```

В строке 4Ø за символом A стоит запятая. Поэтому после вывода величины 1Ø.5 вывод следующей величины может начаться лишь с девятой позиции. Но в первой из функций TAB указано меньшее число 7. Поэтому она игнорируется, и число 1.612 помещается во второй зоне экрана, начиная с девятой позиции. Позиция 21, указанная в скобках во второй функции TAB, остается еще справа, поэтому она выполняется, сдвигая выводимое число в позицию 21.

Функция SPC(X) предписывает процессору вывод X пробелов (X – целое положительное число). В отличие от функции TAB, указывающей конкретную позицию, с которой должен начаться очередной вывод, функция SPC отсчитывает от предыдущего выведенного значения указанное в скобках число позиций. Если в предыдущем примере строку с номером 4Ø заменить следующей

```
4Ø PRINT A,SPC(7);B; SPC(21);C
```

то информация на экране будет расположена таким образом:

```
1Ø.5      1.612
          362.14
```

После вывода величины A, под которую отведена первая зона (восемь позиций), будет пропущено семь позиций, затем выведена величина B, занимающая пять позиций, затем пропущена еще 21 позиция. Так как физическая строка экрана содержит 32 позиции, то неуместившаяся на ней информация

переносится на следующую строку. Величина C будет расположена, начиная с позиции 10 следующей строки экрана.

Кроме функций, применяемых в операторе PRINT, существуют аналогичные самостоятельные операторы HTAB и VTAB.

Оператор HTAB имеет следующий вид:

```
HTAB X
```

где X – целое положительное число.

Выполняя этот оператор, процессор начнет вывод следующего элемента оператора PRINT с X-й позиции экранной строки.

Оператор VTAB имеет вид:

```
VTAB X
```

где X – целое положительное число.

Выполнение этого оператора передвинет курсор на строку экрана с номером X. Следующий за ним оператор PRINT начнет выводить текст с X-й строки.

Операторы HTAB и VTAB значительно расширяют возможности размещения информации на экране. С их помощью можно красиво расположить читаемые страницы текста. Но следует помнить, что в этом случае необходим также оператор HOME (внутренний). С его помощью очищается все поле экрана. Например, фрагмент программы

```
1ØØ HOME
11Ø VTAB 16 : HTAB 9
12Ø PRINT "РАБОТА ОКОНЧЕНА!"
```

очищает экран ВКУ и в центре экрана выводит надпись "РАБОТА ОКОНЧЕНА".

#### Контрольные вопросы

1. Каковы функции оператора PRINT?
2. В каких случаях применяется оператор HOME?
3. Рассказать об операторах передвижения курсора на нужную строку и нужную позицию в строке.
4. Как будет выведен текст на экран ВКУ:

```
1Ø HOME
2Ø VTAB 32: HTAB 4
3Ø RIBBON=2
4Ø PRINT"НАЖМИТЕ НА ЛЮБУЮ КЛАВИШУ"
5Ø VTAB 1 : HTAB 15 : RIBBON = 1
6Ø PRINT "1"
```





**Тема:** Условный переключатель-переход и условный переключатель-вызов, применение их в разветвляющихся алгоритмах.

**Цель работы:** Ознакомление с операторами-переключателями и логическим оператором ветвления, их использование в разветвляющихся алгоритмах.

**Продолжительность работы** - 2 ч.

**Теоретические сведения.** На языке Бейсик возможна еще одна конструкция оператора выбора, - переключатель-переход, имеющий вид:

ON X GOTO N1, N2 [, N3, N4]

где X - любое выражение, принимающее числовое значение; N1-N4 - номера строк. Присутствие номеров, заключенных в квадратные скобки, необязательно.

Выполняя этот оператор, процессор в зависимости от значения X передает управление на одну из указанных после GOTO строк - при X=1 - на N1, при X=2 - на N2, при X=3 - на N3 и т.д. Если значение X меньше единицы или больше числа приведенных строк (меток), то оператор не выполняется и выдается сообщение об ошибке.

Аналогично работает и оператор переключатель-вызов, имеющий вид:

ON X GOSUB N1, N2, N3

где X - любое выражение, принимающее числовое значение; N1-N3 - номера строк, с которых начинаются подпрограммы.

Этот оператор необходим в тех случаях, когда в зависимости от значения X нужно выбрать алгоритм вычисления, реализованный в подпрограммах.

Если в программе необходима работа с логическими выражениями, условный оператор имеет вид:

IF NOT (<логическое выражение>) THEN N

где N - номер строки.

Выполняя его, процессор в случае истинности логического выражения выполнит операторы, находящиеся до строки с номером N. Например:

```
100 IF NOT(A < B OR A ≠ 0) THEN 300
110 A1=A*2-A*B+2*B
120 PRINT A1:GOTO 400
300 A2=0
310 PRINT A2
400 .....
```

Фрагмент программы при A=4, B=5 результатом работы будет иметь: выражение в скобках истинно, следовательно  $A1=2*4-4*5+2*5 = -2$ .

На экран будет выдан результат вычисления: -2.

Затем управление передается строке 400.

### Контрольные вопросы

1. В каких случаях применяется переключатель-переход?
2. В каких случаях применяется переключатель-вызов?
3. Как работает условный оператор с логическими выражениями?
4. Как будет работать фрагмент программы:

100 A=150 : B = 100

300 K=INT  $\left(\frac{360}{A+B} \cdot 0,5 + 3\right)$

310 ON K GOTO 400, 500, 600

.....

400

.....

500

.....

600

.....

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задание 1 или 2 по вариантам;
- оформить отчет.

Отчет по практической работе I4 должен содержать:

- название и цель работы;
- запись общего вида и описание назначения операторов-переключателей и оператора условного перехода с логическим выражением;
- условие задания 1 или 2, блок-схему алгоритма, текст программ на алгоритмическом языке и на языке Бейсик, результат работы программы.

**Задание 1.** Составить программу, реализующую следующий алгоритм:

$$S = \begin{cases} \frac{1}{\sqrt{d}} \operatorname{arctg} \frac{b+c}{\sqrt{d}}, & \text{если } d > 0; \\ \frac{1}{2c} \ln \left| \frac{b-a}{b+c} \right|, & \text{если } d = 0; \\ -\frac{a}{b+c}, & \text{если } d < 0, \end{cases} \quad \text{где } d = ac - b^2.$$

Использовать переключатель-переход.

Задание 2. Составить программу, реализующую следующий алгоритм:

$$y = \begin{cases} a^2 + b^2, & \text{если } x > a/b; \\ a^2 - b^2, & \text{если } a/b \gg x > a^2/b^2. \end{cases}$$

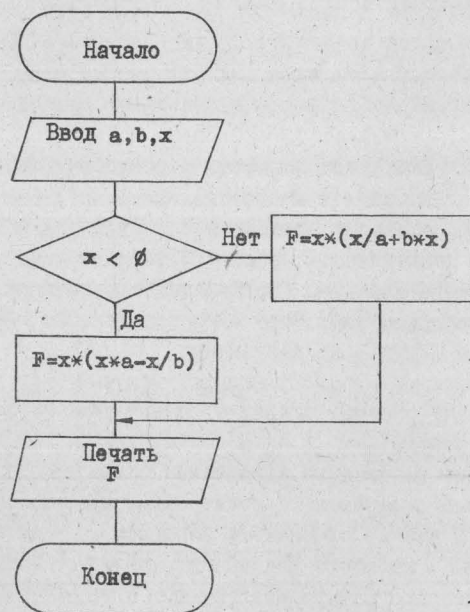
Использовать условный оператор с логическим выражением.

Пример выполнения задания. Составить программу, реализующую следующий алгоритм:

$$F = \begin{cases} x(xa - x/b), & \text{если } x < 0; \\ x(x/a + bx), & \text{если } x \gg 0. \end{cases}$$

Использовать переключатель-вызов.

Блок-схема



Программа на алгоритмическом языке:

```

алг ВЫЗОВ (вещ a, b, x, F)
  арг x
  рез F
нач
  если x < 0
  то F := x * (x * a - x / b)
  иначе F := x * (x / a + b * x)
все
кон
  
```

Программа на языке Бейсик:

```

10 REM ВЫЗОВ
20 PRINT "ВВЕДИТЕ ЗНАЧЕНИЕ А И В": PRINT
30 INPUT A, B
40 PRINT: PRINT "ВВЕДИТЕ X"
50 PRINT: INPUT X
60 IF X < 0 THEN 80
70 K=2 : GOTO 90
80 K=1
90 ON K GOSUB 200, 300
100 PRINT: PRINT "F="; F
110 END
200 F=X*(X*A-X/B)
210 RETURN
300 F=X*(X/A-X*B)
310 RETURN
  
```

Результат работы программы:

```

ВВЕДИТЕ ЗНАЧЕНИЯ А И В
? 10, 20
ВВЕДИТЕ x
? -1
F=9.95
  
```

**Тема:** Составление обучающих и игровых программ.

**Цель работы:** Приобретение навыков составления обучающих и игровых программ.

**Продолжительность работы** – 2 ч.

**Теоретические сведения.** Широкое распространение вычислительной техники обеспечивает возможность обучения с помощью ЭВМ. Это освобождает преподавателя от лишних нагрузок, наглядно демонстрирует материал и одновременно с обучением контролирует усвоение материала. Программы, обеспечивающие эту возможность, называются обучающе-контролирующими.

Существуют также программы, проводящие машинный опрос по определенному курсу или теме и оценивающие знание отвечающего. Эти программы называются контролирующими.

В последние годы увеличилось и количество игровых программ. Игровые программы неоднозначны. Среди них есть класс программ, отрабатывающих быстроту реакции, глазомер и т.п. Такие компьютерные игры можно сравнить с игровыми автоматами.

Можно выделить также класс математических игр, развивающих способность логического мышления, умение предугадать ход противника и выбрать оптимальный вариант хода. Такие игры интересны и взрослым и детям. Они способствуют появлению интереса к математике, кибернетике, программированию; развивают способности, необходимые для решения инженерных задач, проблем управления и экономики. Такие игры более интересны, чем те, в которых необходима лишь быстрота реакции.

В качестве примера обучающе-контролирующей программы рассмотрим программу "Знакомство с клавиатурой". С работой этой программы Вы познакомились в практической работе 1 (задание 1). Текст программы на языке Бейсик приведен в Приложении 5.

Алгоритм обучающе-контролирующих программ состоит из трех основных частей:

- теоретической;
- проверки усвоения материала;
- оценки усвоенных знаний.

Обычно теоретическая часть разбивается на отдельные законченные фрагменты. После изучения каждого фрагмента следует проверка усвоения материала. После полного изучения теоретического материала подводится итог: какое количество ответов было правильным и какое – ошибочным. В зависимости от результата подсчета выдается оценка: следует ли повторить обучение по этой теме или приступить к изучению следующей. Если все темы курса пройдены, выдается сообщение о том, что обучаемый может на практике закрепить изученный материал, что он готов к самостоятельной работе.

Усвоение материала будет лучше, если выделять цветом определенные места текста.

#### Контрольные вопросы

1. Как строится алгоритм обучающе-контролирующей программы?
2. Каким образом можно организовать подсчет неверных ответов?
3. Как создать возможность обучаемому при неверном ответе повторить попытку?
4. Нужно ли увеличить счетчик ошибок при первом неверном ответе, если дана возможность повторного ответа и ответ дан правильно?

Порядок выполнения практической работы:

- ознакомиться с теоретическими сведениями;
- ответить на контрольные вопросы;
- выполнить задание 1 или 2 по вариантам;
- оформить отчет.

Отчет по практической работе 15 должен содержать:

- название и цель работы;
- условие задания 1 или 2;
- блок-схему алгоритма, текст программы на алгоритмическом языке и на языке Бейсик, результат работы программы.

**Задание 1.** Составить игровую программу "Лото с компьютером". На экране ВКУ должны появляться числа от 1 до 99 в произвольном порядке. Использовать функцию случайных чисел `RND(1)`.

После каждой новой фишки, появившейся на экране ВКУ, должна произойти временная задержка, чтобы играющие успели в карточке закрыть выигравшее число. Выигранные числа не должны повторяться. Как только все 99 чисел будут названы, выдается сообщение: "МЕШОК ПУСТОЙ", и программа заканчивает свою работу.

Если выигравшее число имеет название, принятое в игре "Лото", то выводится его название, в противном случае – цифра.

Числа, имеющие названия:

7 - кочерга	69 - туда и обратно
11 - барабанные палочки	77 - топорики
12 - дюжина	80 - бабочки
13 - чертова дюжина	90 - дец
22 - уточки	

Результат работы программы должен быть следующего вида, например:

1-Я ФИШКА - ДЮЖИНА  
2-Я ФИШКА - 5  
и т.д.

**Задание 2.** Составить обучающе-контролирующую программу по теме практической работы 9: "Обработка табличной информации на ЭВМ". Теоретические сведения должны быть изложены в сжатой форме.

**Пример выполнения задания.** Составить игровую программу "Игра Баше". Алгоритм выигрыша первого игрока:

- 1) взять два предмета;
- 2) второй и последующий ходы делать так, чтобы количество предметов, взятых вместе с соперником за очередной ход, в сумме составляло 4.

Количество имеющихся предметов определяется формулой  $7+4*N1=N$  ( $N1=0, 1, 2, \dots$ ). За один ход каждый игрок может взять один, два или три предмета. Проигравшим считается игрок, взявший последний предмет.

Программа на алгоритмическом языке:

алг игра Баше

нач

взять два предмета

пока осталось больше четырех предметов

нц

дать противнику сделать ход

запомнить число X взятых предметов

взять 4-X предметов

кц

дать противнику сделать ход

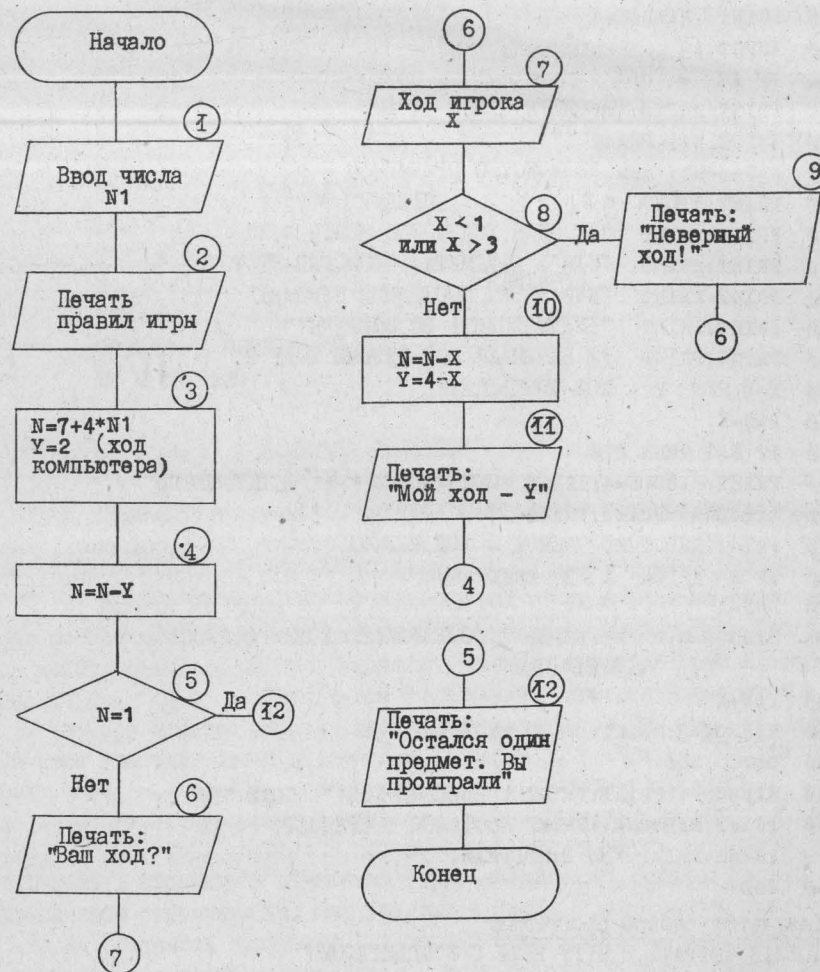
кон

Программа на языке Бейсик:

10 REM ИГРА БАШЕ

20 REM КОМПЬЮТЕР ХОДИТ ПЕРВЫМ И ВЫИГРЫВАЕТ

Блок-схема



```

30 PRINT "ХОТИТЕ СЫГРАТЬ В ИГРУ БАШЕ С КОМПЬЮТЕРОМ?"
35 PRINT: RIBBON=4
40 INPUT AX
50 IF AX = "НЕТ" THEN 280
60 PRINT:PRINT "ВВЕДИТЕ ЦЕЛОЕ ЧИСЛО"
70 INPUT N1: PRINT
80 N=7+4*N1: REM КОЛИЧЕСТВО ПРЕДМЕТОВ - N
90 PRINT "ЕСТЬ - ";N; " _ПРЕДМЕТОВ"
100 PRINT:PRINT "ЗА ОДИН ХОД МОЖНО ВЗЯТЬ 1,2"
110 PRINT:PRINT "ИЛИ 3 ПРЕДМЕТА. ПРОИГРЫВАЕТ ТОТ,"
120 PRINT:PRINT "КТО БЕРЕТ ПОСЛЕДНИЙ ПРЕДМЕТ."
130 PRINT:PRINT "БУДЕМ ХОДИТЬ ПО ОЧЕРЕДИ!"
140 PRINT:PRINT "Я НАЧИНАЮ! МОЙ ПЕРВЫЙ ХОД: 2"
150 Y=2:REM Y - ХОД КОМПЬЮТЕРА
160 N=N-Y
170 IF N=1 THEN 270
180 PRINT:RIBBON=4:PRINT "ОСТАЛОСЬ - ";N; " _ПРЕДМЕТОВ"
190 RIBBON=3:PRINT:PRINT "ВАШ ХОД?"
200 PRINT:INPUT X: REM X - ХОД ИГРОКА
210 IF X < 1 OR X > 3 THEN 260
220 N=N-X
230 RIBBON=4:PRINT:PRINT "ПОСЛЕ ВАШЕГО ХОДА ОСТАЛОСЬ";
N; " _ПРЕДМЕТОВ"
235 Y=4-X
240 RIBBON=2:PRINT:PRINT "МОЙ ХОД:";Y
250 GOTO 160
260 RIBBON=1:PRINT:PRINT "НЕВЕРНЫЙ ХОД!": GOTO 190
270 PRINT:RIBBON=1:PRINT "ОСТАЛСЯ 1 ПРЕДМЕТ"
275 PRINT:PRINT "ВЫ ПРОИГРАЛИ..."
280 END

```

Результат работы программы:

ХОТИТЕ СЫГРАТЬ В ИГРУ БАШЕ С КОМПЬЮТЕРОМ?

?ДА

ВВЕДИТЕ ЦЕЛОЕ ЧИСЛО

? 1

ЕСТЬ - 11 ПРЕДМЕТОВ

ЗА ОДИН ХОД МОЖНО ВЗЯТЬ 1, 2

ИЛИ 3 ПРЕДМЕТА. ПРОИГРЫВАЕТ ТОТ,

КТО БЕРЕТ ПОСЛЕДНИЙ ПРЕДМЕТ.

БУДЕМ ХОДИТЬ ПО ОЧЕРЕДИ!

Я НАЧИНАЮ! МОЙ ПЕРВЫЙ ХОД: 2

ОСТАЛОСЬ - 9 ПРЕДМЕТОВ

ВАШ ХОД?

? 3

ПОСЛЕ ВАШЕГО ХОДА ОСТАЛОСЬ - 6 ПРЕДМЕТОВ

МОЙ ХОД: 1

ОСТАЛОСЬ - 5 ПРЕДМЕТОВ

ВАШ ХОД?

? 2

ПОСЛЕ ВАШЕГО ХОДА ОСТАЛОСЬ - 3 ПРЕДМЕТА

МОЙ ХОД: 2

ОСТАЛСЯ 1 ПРЕДМЕТ

ВЫ ПРОИГРАЛИ...

#### ЗАЧЕТНЫЕ ЗАДАНИЯ

1. Написать программу, которая с указанием цен предлагает покупателю покупать хлеб, молоко, сыр и т.д., запрашивает ввод количества покупаемых товаров и выдает суммарную стоимость покупок.

2. Исходя из того, что 1 января 1984 г. было воскресеньем и год был високосным, составить алгоритм определения дня недели любой даты по действующему календарю. Входными данными являются число К, номер месяца М и год G, а результатом вычислений - номер дня недели N.

3. При осмотре участка леса лесник подсчитал численность деревьев на этом участке: сосен  $n_1$ , елей  $n_2$ , берез  $n_3$ , осин  $n_4$ . Из них здоровых: сосен  $m_1$ , елей  $m_2$ , берез  $m_3$ , осин  $m_4$ . Остальные больны.

Составить программу вычисления: число деревьев на контрольном участке (N), число здоровых деревьев (M), численность различных видов деревьев в процентном отношении ( $P_1$ ), численность больных деревьев в процентном отношении (Q) для участка в целом и для каждого вида ( $Q_1$ ).

4. Составить программу для определения среднего арифметического положительных элементов массивов A и B, используя подпрограмму подсчета суммы.

5. Составить программу вычисления площади круга, который может быть задан радиусом R (признак N=1), диаметром D (признак N=2), длиной окружности L (признак N=3).

$$S = \begin{cases} \pi R^2, & \text{если } N=1; \\ \pi D^2/4, & \text{если } N=2; \\ L^2/4\pi, & \text{если } N=3. \end{cases}$$

Использовать команду ON X GOTO или ON X GOSUB.

6. В текущем месяце бригада в каждый последующий день работы выпускала на три изделия больше, чем в предыдущий день. В первый день было выпущено 42 изделия, что составило 67% дневной нормы. Выполнит ли бригада месячную норму (в месяце 26 дней)? Напечатать: БРИГАДА ВЫПОЛНИЛА НОРМУ" или "БРИГАДА НЕ ВЫПОЛНИЛА НОРМУ" в зависимости от полученного результата.

7. Составить программу вычисления числа сочетаний из N по M. Число сочетаний из N по M определяется по формуле  $N!/M!(N-M)!$ , где N - количество чисел, M - количество элементов перебора. Использовать подпрограмму вычисления факториала.

8. Составить программу, которая в двумерном массиве подсчитывает количество отрицательных, нулевых и положительных элементов, печатает полученные значения и распечатывает все элементы массива в виде матрицы.

9. Составить программу, которая считает годовой процент по вкладам в сберегательной кассе. Общая сумма, годовой процент и количество лет задаются пользователем. Организовать печать в виде:

год	чистая прибыль	всего
-----	----------------	-------

10. В правильной четырехугольной пирамиде SABCD угол между плоскостью основания и боковым ребром равен  $\alpha$ . Составить программу для вычисления:

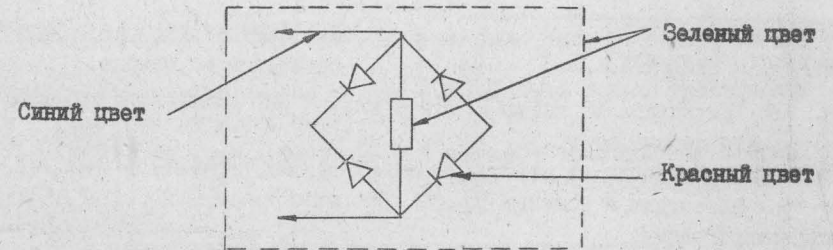
- угла  $\beta$  между боковой гранью и основанием;
- угла  $\gamma$  между двумя пересекающимися боковыми гранями;
- плоского угла  $\nu$  при вершине S.

11. Дан некоторый список жильцов одного подъезда жилого дома с фамилиями и номерами квартир. Определить количество этажей дома по максимальному номеру квартиры в списке. На этаже четыре квартиры. Фамилия в списке занимает 35 позиций, номер квартиры - 3 позиции.

Фамилия	Номер квартиры
Иванов Петр Иванович	064
.....	...

12. Имеются следующие сведения о деталях: наименование (16 символов), год начала выпуска (4 символа), цена (до 999 руб. без копеек), масса (до 99 кг), ширина, длина и высота (до 500 см). Эти сведения нужно ввести в память машины, распечатав только те из них, которые относятся к деталям с годом выпуска не ранее 1967, и посчитав их общую стоимость.

13. Написать программу, которая выводит на печать фрагмент электрической схемы, используя графический режим.



14. Составить программу вычисления периметра  $n$ -угольника по координатам его вершин. Использовать подпрограмму вычисления длины одной стороны.

15. Самолет летит из пункта А в пункт В со средней скоростью V. Составить алгоритм нахождения времени в полете t, если возможны три варианта:

- дует встречный ветер ( $K=1$ );
- ветра нет ( $K=2$ );
- дует попутный ветер ( $K=3$ ),

где K - признак ветра.

Использовать при вычислении команду ON X GOTO или ON X GOSUB.

16. Для покрытия пола размером  $s \times a$  имеются белые и черные облицовочные квадратные плитки со стороной a. Вычислить количество белых и черных плиток, учитывая, что по периметру пола уложен ряд из белых плиток, а внутри плитки уложены в шахматном порядке. Вывести на печать надписи о нужном количестве плиток.

17. Составить программу "Классный журнал". О каждом ученике имеются сведения: фамилия и имя - 35 символов, поле посещения и оценок - 50 символов. Эти сведения вводятся в машину, считается средняя оценка за четверть, количество пропущенных занятий. На печать выдать: фамилию, имя, оценку за четверть, количество пропусков.

18. Дан ряд чисел: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19. Составить программу для вывода на печать двух соседних чисел, первого и второго, второго и третьего, третьего и четвертого и т.д., а также результаты их сложения или умножения, причем при сумме чисел больше 15 выводить на печать их сумму, в противном случае выводить на печать их произведение.

19. Составить программу для перевода градусов в радианы и вычислять  $\sin$ ,  $\cos$ ,  $\operatorname{tg}$  заданного угла. Результат выдать в форме следующей таблицы:

Угол в градусах	Угол в радианах	$\sin$	$\cos$	$\operatorname{tg}$

При  $\alpha > 360^\circ$  и  $\alpha < 0^\circ$  выдать сообщение об ошибке.

20. Вычислить 15 значений суммы  $X=A+B$  при  $A=2,4,6,8,10$ ;  $B=2,4,6$ . Результаты напечатать.

21. Найти один из четырех взаимосвязанных параметров ( $N, I, H, K$ ) по трем известным с помощью формул. Оформить вычисление ( $I + I/100$ ) как подпрограмму.

$$N = \frac{\ln(K/H)}{\ln(1+I/100)}, \quad \text{если } X=1;$$

$$I = (K/H)^{1/N} * 100, \quad \text{если } X=2;$$

$$H = K(1+I/100)^{-N}, \quad \text{если } X=3;$$

$$K = H(1+I/100)^N, \quad \text{если } X=4.$$

Использовать оператор `ON X GOTO`.

22. Расположить элементы массива в порядке уменьшения. Результат напечатать.

23. Найти площадь полной поверхности и объем правильной  $N$ -угольной пирамиды по стороне основания и углу наклона боковой грани к плоскости основания. Пользователем задаются:  $N$  – число сторон,  $A$  – длина стороны,  $B$  – угол в градусах. Сделать проверку входных данных. Должны выполняться следующие условия  $N > 2, A > 0, B < 90^\circ$  и  $B > 0^\circ$ . В противном случае – переход на конец программы.

#### Задания повышенной сложности

24. Составить программу вычисления площади любого выпуклого многоугольника, зная его координаты.

25. Составить программу перевода числа из десятичной системы в шестнадцатеричную, и наоборот.

26. Сформировать из чисел натурального ряда двумерный массив так, чтобы числа находились в порядке возрастания с шагом 1 и располагались в этом массиве спиралью, как показано на рисунке, и вывести матрицу на печать. Количество чисел задается пользователем.

1	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

27. Дан двумерный массив. Сжать его, оставив только неповторяющиеся строки.

28. Дан неупорядоченный двумерный массив с границами  $n$  и  $m$ . Преобразуйте его в линейный массив, упорядоченный по какому-либо признаку.

29. Составить программу поиска и печати простых чисел от 1 до 50. Вывести их на печать.

30. Дан двумерный массив. Подсчитать суммы по строкам и определить строку с наибольшей суммой.

31. Проверить, является ли массив магическим квадратом. Распечатать его элементы в виде матрицы. Квадрат считается магическим, если сумма элементов по строке равна сумме элементов по столбцу и равна сумме элементов по диагоналям. Выдать текстовое сообщение о том, является ли введенный массив магическим квадратом.

32. Составить программу перевода чисел по основанию  $N$  в десятичное число, равное введенному по значению ( $1 < N < 17$ ).

33. Экспертные оценки продукции имеют 10 пунктов. Определить лучшее изделие по сумме экспертных оценок

$$T(I) = A(I) + B(I) + C(I) + D(I) + E(I) + F(I) + G(I) + M(I) + K(I) + L(I).$$

Напечатать результат в порядке расположения от лучших показателей к худшим.



Приложение 1  
ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРАКТИЧЕСКИМ РАБОТАМ

Номер практической работы	Номер вопроса	Ответ
I	4	<p> <span style="border: 1px solid black; padding: 2px;">РУС</span> <span style="border: 1px solid black; padding: 2px;">РЕГ</span> <span style="border: 1px solid black; padding: 2px;">Д</span> <span style="border: 1px solid black; padding: 2px;">И</span> <span style="border: 1px solid black; padding: 2px;">С</span> <span style="border: 1px solid black; padding: 2px;">К</span> <span style="border: 1px solid black; padding: 2px;">Р</span> <span style="border: 1px solid black; padding: 2px;">И</span> <span style="border: 1px solid black; padding: 2px;">М</span> <span style="border: 1px solid black; padding: 2px;">И</span> <span style="border: 1px solid black; padding: 2px;">Н</span> <span style="border: 1px solid black; padding: 2px;">А</span> <span style="border: 1px solid black; padding: 2px;">Н</span> <span style="border: 1px solid black; padding: 2px;">Т</span> <span style="border: 1px solid black; padding: 2px;">РЕГ</span> </p> <p>Одновременно <span style="float: right;">Одновременно</span></p> <p> <span style="border: 1px solid black; padding: 2px;">ЛАТ</span> <span style="border: 1px solid black; padding: 2px;">РЕГ</span> <span style="border: 1px solid black; padding: 2px;">В</span> <span style="border: 1px solid black; padding: 2px;">А</span> <span style="border: 1px solid black; padding: 2px;">2</span> <span style="border: 1px solid black; padding: 2px;">-</span> <span style="border: 1px solid black; padding: 2px;">4</span> <span style="border: 1px solid black; padding: 2px;">РЕГ</span> <span style="border: 1px solid black; padding: 2px;">Ж</span> <span style="border: 1px solid black; padding: 2px;">А</span> <span style="border: 1px solid black; padding: 2px;">РЕГ</span> <span style="border: 1px solid black; padding: 2px;">Ж</span> <span style="border: 1px solid black; padding: 2px;">С</span> </p> <p>Одновременно <span style="margin-left: 100px;">Одновременно</span> <span style="margin-left: 100px;">Одновременно</span></p>
3	3	Удалит строку с номером 110
3	4	Будет выведен на экран ВКУ текст программы, начиная с наименьшего номера строки по строку с номером 110
3	5	Программа будет выполняться со строки с номером 100. Команды с номерами меньше 100 будут выполнены только в том случае, если к ним есть переход из той части программы, номера строк которой больше 100
4	2	Она идет со скоростью - 3 за время - 40 мин
5	4	Запрашивает ввод числа, отвергает ненатуральные числа. Когда натуральное число, благодарит пользователя
6	4	S=55
7	4	II
8	4	КЕТА            ТАРА            РАКЕТА
9	4	86
10	3	Горизонтальная линия зеленого цвета
11	4	Скорость самолета = 372 Скорость поезда = 100

Номер практической работы	Номер вопроса	Ответ
I2	4	На фиолетовом фоне черными буквами будет выведено сообщение: "РАБОТА ОКОНЧЕНА"
I3	4	На строке 1 в центре будет выведена цифра 1 красного цвета, на строке 32 - надпись зеленого цвета, начиная с позиции 4 - "НАЖМИТЕ ЛЮБУЮ КЛАВИШУ"
I4	4	Управление будет передано на строку с номером 600, так как к приняло значение 3

ПРАВИЛА РАБОТЫ УЧАЩИХСЯ В КАБИНЕТЕ  
ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

1. В кабинете вычислительной техники установлена дорогостоящая, сложная и требующая осторожного и аккуратного обращения аппаратура – компьютеры (ПЭВМ), принтер и др. технические средства. Чтобы избежать ее повреждения, спокойно, не торопясь, не задевая столы, входите в кабинет и занимайте отведенное Вам место, ничего не трогая на столах.

2. На Вашем рабочем месте размещены составные части ПЭВМ – системный блок, клавиатура и ВКУ, которые работают под высоким напряжением. Неправильное обращение с аппаратурой, кабелями и ВКУ может привести к тяжелым поражениям электрическим током, вызвать загорание аппаратуры.

Поэтому строго запрещается:

- трогать разъемы соединительных кабелей;
- прикасаться к питающим проводам и устройствам заземления;
- прикасаться к экрану и тыльной стороне ВКУ, клавиатуры;
- включать и отключать аппаратуру без указания преподавателя;
- класть диск, книги, тетради на ВКУ и клавиатуру;
- работать во влажной одежде и влажными руками.

3. При появлении запаха гари немедленно прекратите работу, выключите аппаратуру и сообщите об этом преподавателю, в случае необходимости окажите ему помощь в тушении огня.

4. Перед началом работы:

- убедитесь в отсутствии видимых причин повреждений аппаратуры на рабочем месте;
- сядьте так, чтобы линия зрения приходилась в центр экрана, на расстоянии 60–70 см от него, чтобы, не наклоняясь, работать на клавиатуре и воспринимать передаваемую на экран ВКУ информацию;
- разместите на столе тетрадь и учебное пособие так, чтобы они не мешали работе на ПЭВМ;
- хорошо разберитесь в особенностях применяемого в работе оборудования;
- начинайте работу только по указанию преподавателя.

5. Во время работы:

- строго выполняйте все перечисленные выше правила, а также указания преподавателя; все операции при работе на ПЭВМ выполняйте в стро-

гой последовательности, изложенной преподавателем. При необходимости обращайтесь к преподавателю;

- следите за исправностью аппаратуры и немедленно прекращайте работу при появлении необычного звука или самопроизвольного отключения аппаратуры. Немедленно докладывайте об этом преподавателю;
  - плавно нажимайте на клавиши, не допуская резких ударов;
  - не пользуйтесь клавиатурой, если не подключено напряжение;
  - работайте на клавиатуре чистыми руками;
  - никогда не пытайтесь самостоятельно устранять неисправности в работе аппаратуры;
  - не вставайте со своих мест, когда в кабинет входят посетители.
6. Работа на ПЭВМ требует большого внимания, четких действий и самоконтроля.

Поэтому нельзя работать:

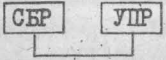

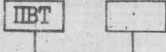
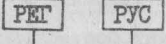
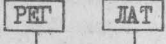
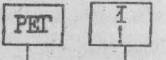
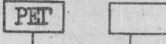
- при недостаточном освещении;
- при плохом самочувствии (в этом случае надо обратиться к врачу).






7. Вы должны хорошо знать и строго придерживаться настоящих правил, точно следовать указаниям преподавателя, чтобы:

- избежать несчастных случаев;
- успешно овладеть знаниями, умениями, навыками;
- сберечь государственное имущество – вычислительную технику и оборудование.

Вы отвечаете за состояние рабочего места и сохранность размещенного на нем оборудования. Невыполнение правил – грубейшее нарушение порядка и дисциплины.

КЛАВИШИ УПРАВЛЕНИЯ

Обозначение клавиши	Действие, вызываемое нажатием клавиши
 Одновременно	<p>Прекращение выполнения любой программы и вывод ПЭВМ в определенное начальное состояние (обычно, в состояние приглашения к диалогу, когда ПЭВМ ждет директив от пользователя, высвечивая слева на экране знак " ] " и мигающий курсор)</p> <p>Ввод директив, набранных на клавиатуре, после чего ПЭВМ воспринимает набранную директиву и исполняет ее</p> <p>Повторная выдача символа, нажатого одновременно с клавишей ПБТ. Действие прекращается, когда отпускается одна из нажатых клавиш</p>
	
 Одновременно	
 Одновременно	
 Одновременно	
 Одновременно	
 Одновременно	<p>Переключение в русский набор букв</p> <p>Переключение в латинский набор букв</p> <p>Выдача на экран ВКУ знака "!" (нажатие одновременно с клавишей РЕГ любой из клавиш, содержащих цифры и специальные знаки, выдаст символ, находящийся на <u>нижнем</u> регистре)</p> <p>При фиксированном русском регистре выдаст символ на латинском регистре, соответствующий нажатой клавише, а при фиксированном латинском регистре – символ на русском регистре, соответствующий нажатой клавише</p>

Обозначение клавиши	Действие, вызываемое нажатием клавиши
	Ввод в режим редактирования. Подробнее о работе клавиши см. в практической работе 1
	Перемещение курсора на одну строку вниз
	Перемещение курсора на одну строку вверх
	Перемещение курсора на одну позицию вправо
	Перемещение курсора на одну позицию влево

СООБЩЕНИЯ ОБ ОШИБКАХ, ВОЗМОЖНЫХ ПРИ ОТЛАДКЕ ПРОГРАММ

Сообщение, выводимое на экран ВКУ	Пояснение выводимого сообщения
NO FOR ERROR	NEXT без FOR
SYNTAX ERROR	Синтаксическая ошибка
NO GOSUB ERROR	RETURN без GOSUB
NO DATA ERROR	Мало данных
ILLEGAL VALUE ERROR	Ошибочное значение
OVERFLOW ERROR	Переполнение
OUT OF MEMORY ERROR	Мало памяти
UNDEF STATEMENT ERROR	Нет номера
SUBSCRIPT ERROR	Ошибочный индекс
REDIM ARRAY ERROR	Ошибка в размере массива
DIVISION ZERO	Деление на нуль
TYPE ERROR	Ошибочный тип
LONG STRING ERROR	Строка длинна
UNDEF NAME ERROR	Нет метки
BYTE UNCOMPL ERROR	Байт неполон
LABEL ERROR	Ошибочная метка
OPCODE ERROR	Ошибочный код операции
DOUBLE DEF NAME ERROR	Повторение определяемого имени

Текст программы "Знакомство с клавиатурой"

```

JLOAD KONTRC
JLIST
1 HOME : VTAB 16
3 RIBBON= 3
4 PRINT "* СЕВАСТОПОЛЬСКИ
  й РАЙОН,УПК-2 *"
5 FOR I = 1 TO 50: Z13 =
  COS (8): NEXT I
8 HOME
9 DIM A$(8)
10 RIBBON= 2
15 VTAB 16
20 RESTORE : PRINT : PRINT
  "*** ЗНАКОМСТВО С КЛА
  ВИАТУРОЙ ***"
25 FOR I = 1 TO 78: Z13 =
  COS (8): NEXT I
26 HOME
30 PRINT
40 DATA "1987", "0.876", "
  BASIC", "PRINT-ПЕЧАТЬ"
  , "***ИНФОРМАТИК***", "
  COMPUTER-КОМПЬЮТЕР", "
  10 КЛАСС", "Y=15X-6U/(
  35+67V)*8R"
50 FOR I = 1 TO 8
60 READ A$(I)
70 NEXT I
80 LET C$ = " НАБЕРИТЕ
  : "
90 LET U = 1: REM U-СЧЕТЧ
  ИК ОШИБОК
100 FOR I = 1 TO 8
110 PRINT
115 RIBBON= 2
120 PRINT C$;A$(I)
125 PRINT
130 INPUT X$
140 IF X$ = A$(I) GOTO 14
  8
141 U = U + 1
142 PRINT
143 RIBBON= 1: PRINT "НЕП
  РАВИЛЬНО! ПРОВЕРЬТЕ,"
  : PRINT : PRINT "КАКИ
  М АЛФАВИТОМ И РЕГИСТР
  ОМ": PRINT : PRINT "В
  Ы ПОЛЬЗУЕТЕСЬ!"
144 GOTO 110
    
```

```
148 PRINT
149 PRINT "ПРАВИЛЬНО!"
150 NEXT I
160 IF U > 3 GOTO 220
170 ON U GOTO 189,199,209
```

```
180 PRINT
189 PRINT
190 RIBBON= 3: PRINT "ОТЛ
ИЧНО!": PRINT : GOTO
230
199 PRINT
200 RIBBON= 4: PRINT "ХОР
ОМО...": PRINT : GOTO
```

```
230
209 PRINT
210 RIBBON= 5: PRINT "ДА.
..ТОЛЬКО УДОВЛЕТВОРИТ
ЕЛЬНО!": PRINT : PRINT
"ПОРАБОТАЙТЕ ЕЩЕ РАЗО
К!": PRINT : GOTO 10
220 RIBBON= 6: PRINT : PRINT
"ПЛОХО,НО НЕ БЕЗНАДЕЖ
НО!": PRINT : PRINT "
БУДЬТЕ БОЛЕЕ ВНИМАТЕЛ
ЬНЫ": PRINT : PRINT "
ПРИ РАБОТЕ С КОМПЬЮТЕ
РОМ!": GOTO 10
230 PRINT "ВЫ ГОТОВЫ К РА
БОТЕ": PRINT : PRINT
"С ПРОСТЫМИ ПРОГРАММА
МИ": PRINT : PRINT "Н
А ЯЗЫКЕ БЕЙСИК-АГАТ!"
240 END
```

## ЛИТЕРАТУРА

1. Основы информатики и вычислительной техники. Программа для средних учебных заведений. - М.: Просвещение, 1985.
2. Основы информатики и вычислительной техники. Пробное учебное пособие для средних учебных заведений. Ч. I /Под ред. А.П.Ершова, В.М.Монахова. - М.: Просвещение, 1985.
3. Основы информатики и вычислительной техники. Пробное учебное пособие для средних учебных заведений. Ч. II /Под ред. А.П.Ершова, В.М. Монахова. - М.: Просвещение, 1986.

ОГЛАВЛЕНИЕ

Общие положения .....	3
Методические рекомендации по проведению практических занятий .....	4
Практические занятия .....	6
Практическая работа 1. Персональная ЭВМ .....	6
Практическая работа 2. Общие сведения о языке Бейсик .....	13
Практическая работа 3. Простейшие программы, реализующие линейный алгоритм на языке Бейсик, выполнение их на ПЭВМ .....	17
Практическая работа 4. Операторы ввода и вывода в языке Бейсик. Выполнение операций ввода-вывода на ПЭВМ .....	21
Практическая работа 5. Разветвляющийся алгоритм (оператор выбора). Запись разветвляющегося алгоритма на языке Бейсик, выполнение его на ПЭВМ .....	25
Практическая работа 6. Циклический алгоритм (оператор повторения с параметром). Запись циклического алгоритма на языке Бейсик, выполнение его на ПЭВМ .....	30
Практическая работа 7. Вспомогательные алгоритмы вычисления значений функций, выполнение их на ПЭВМ .....	34
Практическая работа 8. Работа с литерными величинами. Функции обработки литерных величин в языке Бейсик.....	39
Практическая работа 9. Обработка табличной информации на ПЭВМ .....	44
Практическая работа 10. Обработка графической информации на ПЭВМ .....	48
Практическая работа 11. Запись вспомогательных алгоритмов в виде подпрограммы, выполнение их на ПЭВМ .....	53
Практическая работа 12. Режимы вывода текстов на экран ВКУ ПЭВМ .....	57

Практическая работа 13. Операторы размещения информации на экране ВКУ .....	59
Практическая работа 14. Условный переключатель-переход и условный переключатель-вызов, применение их в разветвляющихся алгоритмах .....	64
Практическая работа 15. Составление обучающих и игровых программ .....	68
Зачетные задания .....	73

Приложения:

1. Ответы на контрольные вопросы к практическим работам.....	78
2. Правила работы учащихся в кабинете вычислительной техники .....	80
3. Клавиши управления .....	82
4. Сообщения об ошибках, возможных при отладке программ..	84
5. Текст программы "Знакомство с клавиатурой" .....	85
Литература .....	87

Ответственные за выпуск:

Г.Н.Третьякова, Е.В.Протасова, Л.В.Трофимова

Подписано в печать 12.07.88. Формат 60x84<sup>1</sup>/16. Заказ 570  
 Объем п.л. 5,5. Уч.-изд.л.4,3. Тираж 320 экз. Цена 22 коп.

ВМНУЦ ВТИ. 119034. Москва, Кропоткинская, 40

Типография Статуправления Московской области, Москва,  
 Старопанский пер.,3