

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Кафедра программного обеспечения информационных технологий

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

для студентов специальности 40 01 01

“Программное обеспечение информационных технологий”
дневной формы обучения

В 4-х частях

Часть 1

Минск 2004

УДК 004.41 (075.8)
ББК 32.973-018.2 я 73
О-75

Р е ц е н з е н т:
заведующий кафедрой информатики Минского государственного высшего
радиотехнического колледжа,
канд. техн. наук, доц. Ю.А.Скудняков

А в т о р ы:
Л.А. Глухова, Е.П. Фадеева, Е.Е. Фадеева, С.В. Болтак

Основы алгоритмизации и программирования: Лаб. практикум для
О-75 студ. спец. 40 01 01 “Программное обеспечение информационных
технологий” дневной формы обуч. В 4 ч. Ч.1 / Л.А.Глухова,
Е.П.Фадеева, Е.Е.Фадеева, С.В.Болтак. – Мн.: БГУИР, 2004. – 42 с.
ISBN 985-444-614-X (ч.1)

В первой части лабораторного практикума приведены материалы для выполнения лабораторной работы № 1. Дано описание среды программирования Borland Pascal 7.0. Рассмотрены правила разработки и отладки программы, написанной на языке Pascal, на примере циклической программы с известным числом повторений. Приведены варианты индивидуальных заданий по лабораторной работе № 1.

УДК 004.41 (075.8)
ББК 32.973-018.2 я 73

ISBN 985-444-614-X (ч.1)
ISBN 985-444-616-6

© Коллектив авторов, 2004
© БГУИР, 2004

СОДЕРЖАНИЕ

Лабораторная работа № 1	4
1. Среда Borland Pascal 7.0	4
2. Пример разработки и отладки программы.....	9
3. Варианты индивидуальных заданий по лабораторной работе № 1	16
3.1. Первая группа заданий	16
3.1.1. Варианты индивидуальных заданий	16
3.1.2. Пример выполнения индивидуального задания	21
3.2. Вторая группа заданий	32
3.3. Задания для самостоятельной работы.....	34
Литература	41

ЛАБОРАТОРНАЯ РАБОТА № 1

Цель работы. Изучение среды Borland Pascal 7.0. Изучение структуры и правил записи программ на языке Borland Pascal. Приобретение навыков разработки, отладки и выполнения программ в среде Borland Pascal 7.0 на примере программирования циклических алгоритмов.

1. Среда Borland Pascal 7.0

Система программирования Borland Pascal 7.0 представляет собой интегрированную среду, включающую экранный редактор; компилятор; редактор связей; отладчик. Интегрированная среда позволяет набирать тексты программ с использованием встроенного редактора текстов, компилировать их, выполнять, проводить отладку программ.

Для запуска Pascal из среды Windows необходимо в меню Start выбрать подменю Programs, переместить указатель мыши на строку Borland Pascal и в выпавшем подменю опять выбрать строку Borland Pascal.

После успешного вызова системы появляется экранный редактор, имеющий вид, приведенный на рис.1.



Рис.1. Экранный редактор

Верхняя строка экрана содержит главное меню, нижняя – функциональные клавиши и краткую справку об их назначении.

Вся остальная часть экрана принадлежит окну редактора, предназначенному для ввода и корректировки исходного текста программы.

Имя дискового файла, откуда был прочитан текст программы (новому файлу присваивается имя NONAME00.PAS), располагается сверху посередине.

Сверху по краям размещаются два специальных поля, используемых при работе с устройством ввода «мышь». Квадрат слева позволяет закрыть текущий файл, стрелка справа – развернуть на весь экран или минимизировать окно текущего редактируемого файла.

Слева от стрелки размещается цифра 1 – номер окна. В Borland Pascal можно работать одновременно с несколькими программами, каждая из которых будет размещена в отдельном окне редактора.

Внизу слева два числа, записанные через “:”, указывают текущую

позицию курсора - номер строки и позицию в строке соответственно.

Кроме окна редактора, в Borland Pascal используются также окна отладочного режима, вывода результатов работы программы, справочной службы, стека, регистров.

Главное меню можно активизировать с помощью мыши или нажатия клавиши F10. Такая клавиша (или комбинация клавиш) называется *горячей клавишей*.

Главное меню включает в себя следующие пункты:

- **File** – действия с файлами и выход из системы;
- **Edit** – редактирование текста;
- **Search** – поиск текстовой строки, процедуры, функции или места ошибки;
- **Run** – выполнение программы;
- **Compile** – компиляция программы из активного окна;
- **Debug** – отладка программы;
- **Tools** – запуск вспомогательных программ;
- **Options** – установка параметров среды;
- **Window** – работа с окнами;
- **Help** – справочная служба.

Главное меню, по своей сути, является лишь оглавлением для тех подменю, которые включает в себя каждый его пункт.

Пункт File содержит следующие подменю:

- **New** – создает и открывает новое окно редактора с именем *nonameAA.pas*, где AA порядковый номер окна с именем *noname*;
- **Open** – открывает для редактирования ранее созданный и сохраненный файл;
- **Save** – записывает содержимое активного окна редактора в файл; горячая клавиша F2;
- **Save as** – записывает содержимое активного окна редактора в файл под другим именем;
- **Save all** – записывает содержимое всех окон редактора в соответствующие файлы;
- **Change dir** – изменяет текущий каталог пользователя на вновь введенный;
- **Print** – печатает текущий файл;
- **Print setup** – настраивает среду для печати текущего файла;
- **Dos shell** – осуществляет временный выход в MS DOS. Для возврата в редактор среды Pascal необходимо в командной строке набрать *Exit*;
- **Exit** – завершает работу в среде Borland Pascal; горячая клавиша Alt-X.

Пункт Edit содержит подменю:

- **Undo** – отменяет последнее изменение, т.е. восстанавливает вид активного окна редактора, предшествующий данному изменению; горячая клавиша Alt-Backspace;
- **Redo** – отменяет действие последней команды **Undo**;
- **Cut** – удаляет выделенный блок, помещая его при этом в буфер обмена Clipboard; горячая клавиша Shift-Del;
- **Copy** – копирует выделенный блок в буфер обмена Clipboard; горячая клавиша Ctrl-Ins либо Ctrl-C;
- **Paste** – копирует содержимое буфера обмена Clipboard в активное окно редактора; горячая клавиша Shift-Ins либо Ctrl-V;
- **Clear** – удаляет выделенный блок, не помещая его в буфер обмена Clipboard; горячая клавиша Ctrl-Del;
- **Show Clipboard** – показывает содержимое буфера обмена.

Пункт Search включает в себя подменю:

- **Find** – осуществляет поиск текстовой строки в активном окне редактора;
- **Replace** – отыскивает в активном окне заданный текстовый фрагмент и заменяет его на вновь введенный;
- **Search Again** – повторяет поиск (и замену) заданного фрагмента во всем тексте;
- **Go To Line Number** – перемещает курсор в окне редактора на строку с заданным номером;
- **Show Last Compiler Error** – выделяет строку текста с обнаруженной при последней компиляции синтаксической ошибкой;
- **Find Error** – осуществляет поиск строки в тексте программы, вызвавшей ошибку выполнения;
- **Find Procedure** – осуществляет поиск в тексте программы заданной процедуры или функции.

Пункт Run содержит подменю:

- **Run** – осуществляет компиляцию, компоновку и выполнение программы из файла редактора; горячая клавиша Ctrl-F9;
- **Step Over** – осуществляет пошаговое выполнение головной программы, исключая пошаговое выполнение процедур и функций пользователя; горячая клавиша F8;
- **Trace Into** – обеспечивает пошаговое выполнение программ, включая пошаговое выполнение всех процедур и функций, написанных пользователем; горячая клавиша F7;

— **Go To Cursor** – осуществляет выполнение программы после компиляции и компоновки до первого оператора в выделенной строке; горячая клавиша F4;

— **Program Reset** – прекращает отладку программы и удаляет выполняющуюся программу из памяти, закрывая все открытые в ней на данный момент файлы; горячая клавиша Ctrl-F2;

— **Parameters** – позволяет задать строку параметров, которые передаются выполняемой программе.

Пункт Compile содержит подменю:

— **Compile** – компилирует программу или модуль, находящийся в активном окне редактора; горячая клавиша Alt-F9;

— **Make** – компилирует программу с перекомпиляцией лишь измененных модулей (TPU-файлов); горячая клавиша F9;

— **Build** – полностью перекомпилирует программу, включая перекомпиляцию всех модулей;

— **Target** – позволяет выбрать режим компиляции;

— **Primary File** – задает имя начального файла, т.е. того файла, с которого необходимо начинать компиляцию и выполнение программы;

— **Clear Primary File** – очищает имя, заданное опцией **Primary File**;

— **Information** – показывает статистику программы.

Пункт Debug включает в себя:

— **Breakpoints** – позволяет просматривать, корректировать и удалять контрольные точки;

— **Call Stack** – активизирует окно программного стека, отображающее все вызовы процедур и функций; горячая клавиша Ctrl-F3;

— **Register** – активизирует окно регистра, отображающее текущее состояние всех регистров микропроцессора ПК;

— **Watch** – активизирует окно отладки;

— **Output** – активизирует окно программы;

— **User Screen** – активизирует окно программы и распакивает его на весь экран; горячая клавиша Alt-F5;

— **Evaluate / Modify** – позволяет в процессе отладки просмотреть или заменить значение любой переменной, а также найти значение любого выражения; горячая клавиша Ctrl-F4;

— **Add Watch** – добавляет в окно наблюдения (отладки) переменные или выражения вместе с их текущими значениями, тем самым позволяя наблюдать за изменением этих значений; горячая клавиша Ctrl-F7;

— **Add Breakpoint** – позволяет установить в текущей строке контрольную точку; горячая клавиша Ctrl-F8 позволяет как устанавливать, так и снимать контрольную точку.

Пункт Tools содержит подменю:

- **Messages** – активизирует окно сообщений. Среда ищет и показывает файл с нужным фрагментом текста программы;
- **Go To Next** – ищет фрагмент, заданный следующим сообщением в окне **Messages**; горячая клавиша Alt-F8;
- **Go To Previous** – ищет фрагмент, соответствующий предыдущему сообщению в окне **Messages**; горячая клавиша Alt-F7;
- **Grep** – инициирует работу утилиты **Grep**; по умолчанию ищет во всех файлах с расширением .PAS текущего каталога имя переменной, процедуры или функции, на которой к моменту вызова опции стоял курсор; горячая клавиша Shift-F2.

Пункт Options включает в себя:

- **Compiler** – задает параметры, с помощью которых можно управлять генерацией машинного кода программы;
- **Memory Sizes** – регулирует размер памяти, которую занимает рабочая программа;
- **Linker** – регулирует режим работы компоновщика;
- **Debugger** – определяет используемый отладчик и режим обновления экрана дисплея в процессе отладки;
- **Directories** – определяет группы функциональных каталогов: *EXE&TPU directories* указывает каталог, где будут располагаться готовые к работе .exe- и .tpu-файлы; *Include directories* содержит имена каталогов, в которых необходимо искать включаемые файлы, если таковые не найдены в текущем каталоге; *Unit directories* содержит каталоги, в которых будет вестись поиск .tpu-файлов, если среда не обнаружит их в текущем каталоге; *Object directories* указывает имена каталогов для поиска .obj-файлов, если таковые не найдены в текущем каталоге;
- **Environment** – позволяет установить некоторые дополнительные опции окружения (цвет, работа с мышью и т.д.);
- **Open** – позволяет задать имя конфигурационного файла, где содержится информация о настройке среды;
- **Save** – сохраняет текущую настройку среды в конфигурационном файле;
- **Save As** – позволяет указывать имя каталога и файла, в котором среда будет сохранять свою настройку.

Пункт Window содержит подменю:

- **Tile** – располагает окна так, чтобы каждое было видно на экране, уравнивая, по возможности, их размеры;
- **Cascade** – располагает окна на экране в каскадном режиме;
- **Close All** – закрывает все открытые окна;

- **Refresh Display** – обновляет экран путем удаления следов программы, работавшей в режиме отладки;
- **Size / Move** – позволяет перемещать активное окно по экрану либо изменять его размеры; горячая клавиша Ctrl-F5;
- **Zoom** – распаивает окно на весь экран или возвращает ему прежний вид; горячая клавиша F5;
- **Next** – активизирует следующее окно; горячая клавиша F6;
- **Previous** – активизирует предыдущее окно; горячая клавиша Shift-F6;
- **Close** – закрывает активное окно; горячая клавиша Alt-F3;
- **List** – выводит список всех открытых окон; горячая клавиша Alt-0.

Пункт **Help** содержит подменю:

- **Contents** – позволяет просматривать содержимое справочной службы;
- **Index** – выводит алфавитный список всех ссылок справочной службы; горячая клавиша Shift-F1;
- **Topic Search** – выдает информацию, обнаружив зарезервированное слово в окрестностях курсора; горячая клавиша Ctrl-F1;
- **Previous Topic** – выводит предыдущее справочное сообщение; горячая клавиша Alt-F1;
- **Using Help** – дает разъяснения о том, как пользоваться справочной службой;
- **Files** – позволяет установить нужный файл справочной службы;
- **Compiler Directives** – выводит справку директив компилятора;
- **Procedures And Functions** – содержит справку о стандартных процедурах и функциях;
- **Reserved Words** – содержит справку о зарезервированных словах;
- **Standard Units** – содержит справку о стандартных модулях;
- **Borland Pascal Language** – содержит справку о языке Borland Pascal;
- **Error Messages** – содержит справку сообщений об ошибках;
- **About** – содержит информацию об авторских правах.

2. Пример разработки и отладки программы

Рассмотрим простейший пример разработки алгоритма, написания и отладки программы.

Пример. Для параметра A , изменяющегося от 100 до 1000 с шагом 10, и параметра B , изменяющегося от -10 до 10 с шагом 1, вычислить значение функции $Y=A/B$.

Очевидно, что для решения данной задачи необходимо использовать циклический процесс. В условии даны начальные значения переменных A и B и законы, по которым они изменяются. Таким образом, в алгоритме необходимо

предусмотреть два цикла, один из которых должен быть вложен в другой. Следовательно, степень вложенности циклов равна двум.

Для решения данной задачи можно реализовать два равноценных варианта алгоритма.

Вариант №1. Во внешнем цикле изменяется переменная A , во внутреннем – переменная B .

Вариант №2. Во внешнем цикле изменяется переменная B , во внутреннем – переменная A .

Схема алгоритма для варианта №1 представлена на рис.2.

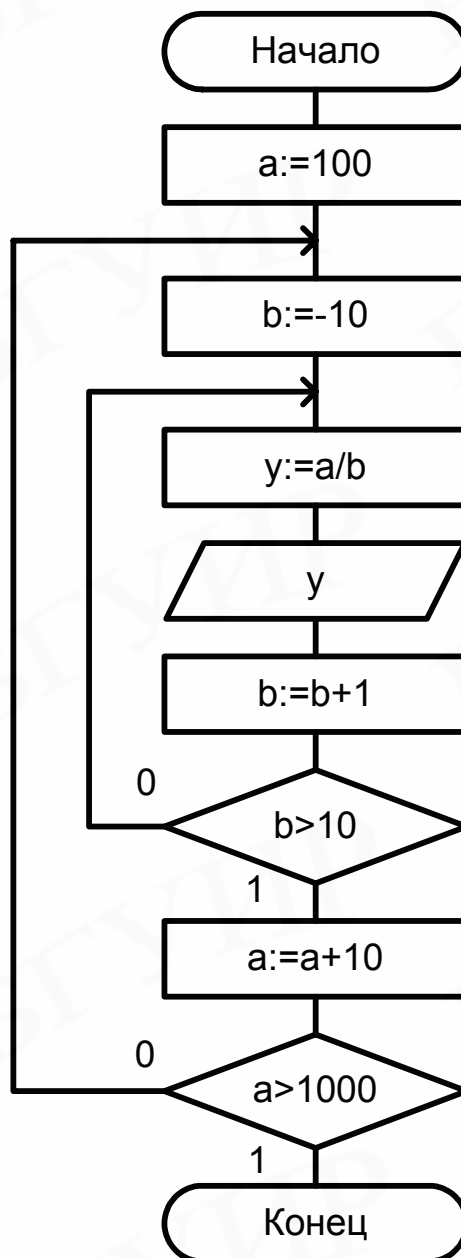


Рис.2. Схема алгоритма

Соответствующая программа на языке Pascal имеет следующий вид:

```

{ 1} Program P1;
{ 2} Uses Crt;
{ 3} Var a, b, y: Integer;
{ 4} Begin
{ 5}     a:= 100;
{ 6}     b:= -10;
{ 7}     Repeat
{ 8}         Repeat
{ 9}             y:= a/b; b:= b + 1; Writeln ( 'a =', a:5, 'b =', b:5, 'y =', y:6:2);
{10}             Delay(500);
{11}         Until b > 10;
{12}         a := a + 10;
{13}     Until a > 1000;
{14} End.

```

В данной программе намеренно допущен ряд ошибок, чтобы продемонстрировать некоторые принципы отладки программы. Читателю предлагается попытаться самостоятельно их отыскать. Процесс отладки программы описывается ниже.

Итак, при запуске программы P1 на компиляцию будет выявлена первая ошибка (рис.3).

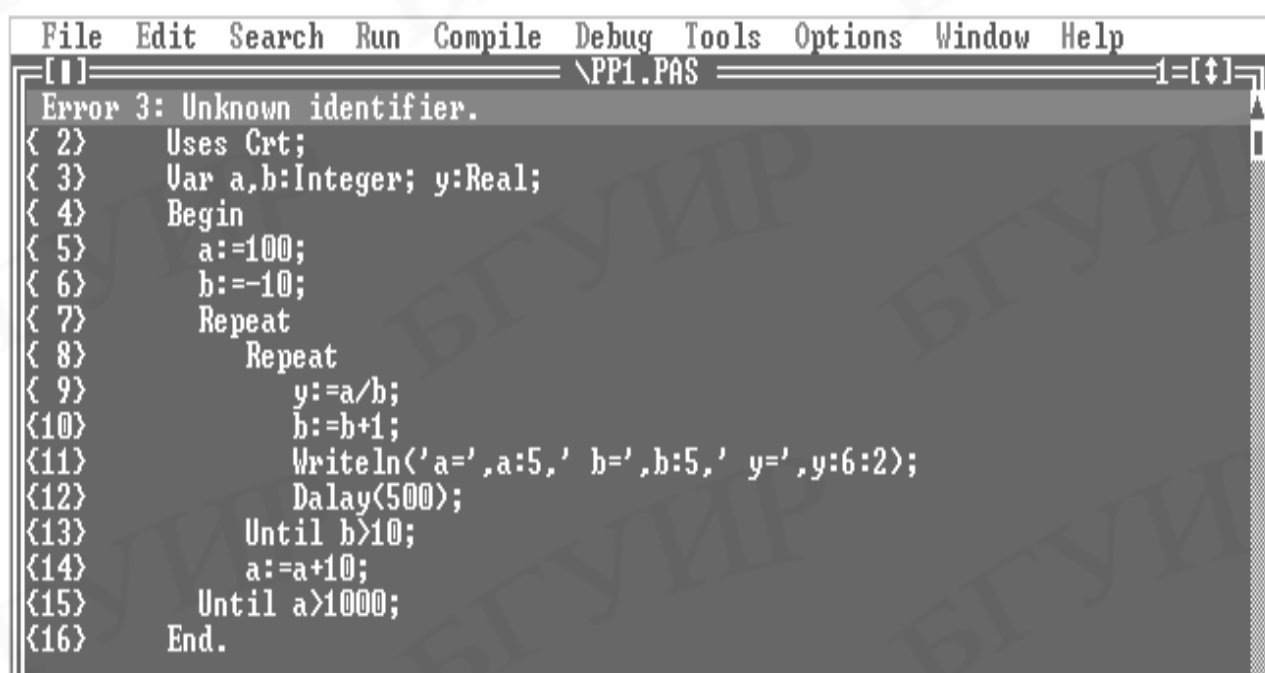


Рис.3. Первый шаг отладки программы

Ошибка обнаружена в 9-й строке, о чем свидетельствует мигающий в этой строке курсор. Код ошибки трансляции 26 “Несоответствие типа” свидетельствует о том, что тип результата операции не соответствует типу переменной, в которую заносится данный результат. Однако 9-я строка содержит три оператора. Для уточнения, в каком конкретно операторе

допущена ошибка, необходимо расположить каждый оператор на отдельной строке. В результате при следующем запуске программы на компиляцию станет ясно, что ошибка обнаружена в операторе $y := a / b$. Ошибка возникает, так как операция деления независимо от типа операндов всегда возвращает вещественный тип результата, и, следовательно, результат операции деления не может быть присвоен целочисленной переменной. Для переменной y необходимо определить тип *Real*.

После исправления этой ошибки при очередном запуске программы на компиляцию возникает следующая ошибка. Код ошибки трансляции 3 “Неизвестный идентификатор” приведен на рис.4.



```
File Edit Search Run Compile Debug Tools Options Window Help
\PP1.PAS
Error 3: Unknown identifier.
< 2> Uses Crt;
< 3> Var a,b:Integer; y:Real;
< 4> Begin
< 5>     a:=100;
< 6>     b:=-10;
< 7>     Repeat
< 8>         Repeat
< 9>             y:=a/b;
<10>             b:=b+1;
<11>             WriteLn('a=',a:5,' b=',b:5,' y=',y:6:2);
<12>             Dalay(500);
<13>         Until b>10;
<14>         a:=a+10;
<15>     Until a>1000;
<16> End.
```

Рис.4. Второй шаг отладки программы

Ошибка возникла в 12-й строке, где была сделана попытка произвести задержку работы программы, воспользовавшись стандартной процедурой Borland Pascal. Однако при написании имени процедуры была допущена орфографическая ошибка. Для проверки данного предположения необходимо вызвать в меню **Help** подменю **Procedures And Functions** (рис.5), в котором следует найти необходимую процедуру.

После исправления данной ошибки (имени Dalay на Delay) необходимо вновь запустить программу на компиляцию. На этот раз компиляция прошла успешно. Однако при запуске программы на выполнение возникла новая ситуация. Код ошибки выполнения 200 “Деление на нуль” (рис.6).

Для локализации ошибки необходимо использовать пошаговый режим. Для этого в меню **Debug** следует выбрать подменю **Add Watch** и добавить в окно отладки переменные a , b , y , так как, скорее всего, их изменение ведет к возникновению ошибочной ситуации (рис.7). С помощью нажатия клавиши F8

необходимо пошагово выполнять программу до тех пор, пока не станет очевидным, что в определенный момент делитель b принимает значение 0.

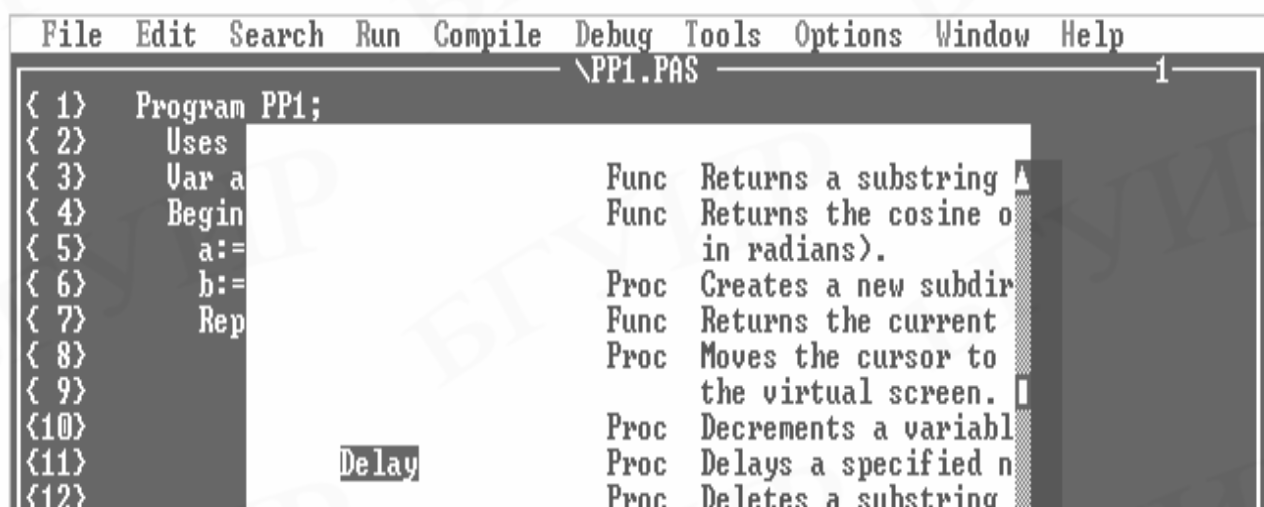


Рис.5. Поиск имени процедуры в меню Help

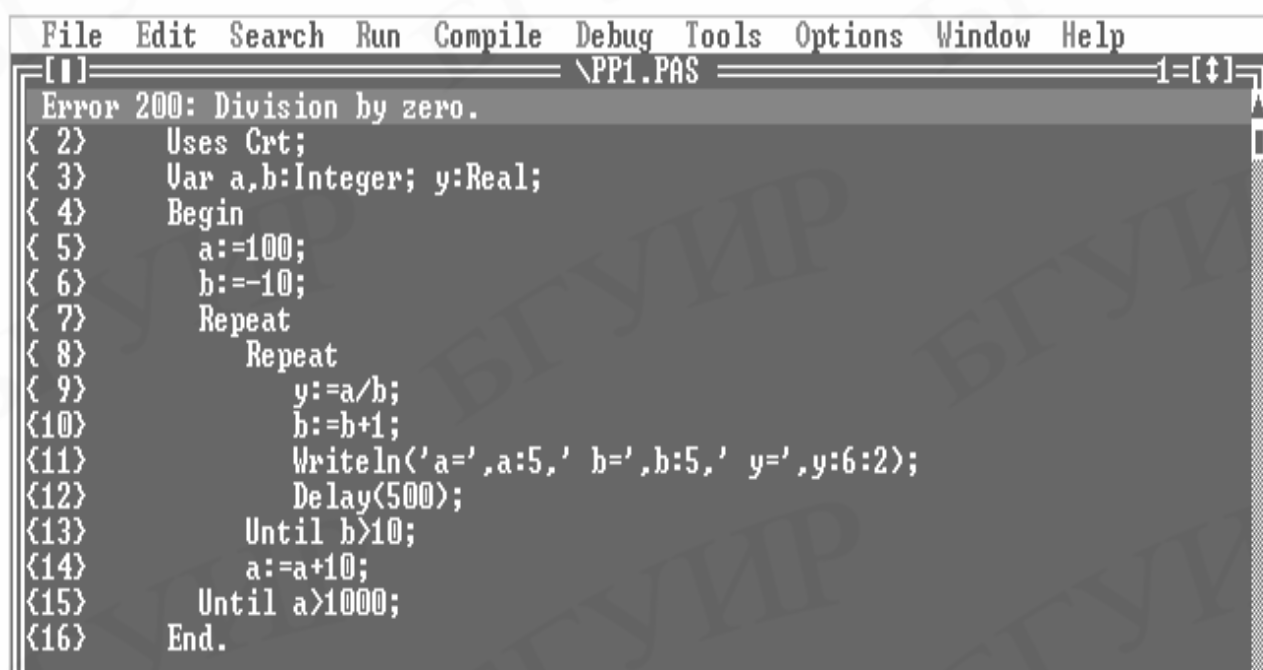


Рис.6. Ошибка деления на нуль

Это значит, что при значении переменной $b=0$ функция a/b не определена. В программу необходимо вставить проверку деления на нуль и вновь произвести пошаговое выполнение программы (рис.8).

В ходе пошагового выполнения программы становится очевидным, что наращивание переменной b прекращается после того, как она станет равной 0. Чтобы избежать данной ситуации, оператор $b:=b+1$; необходимо вынести за

пределы операторных скобок *Begin...End* и разместить после строки №15 программы (рис.9).

В результате данных исправлений программа выполняется, но результаты ее выполнения не соответствуют предполагаемым, так как значение переменной *y* не изменяется и равно 10 для всех значений переменных *a* и *b* :

```
a= 800 b= 80 y= 10.00
a= 810 b= 81 y= 10.00
.....
a= 1000 b= 100 y= 10.00
```

Для выявления причины данного несоответствия следует вновь воспользоваться пошаговым режимом отладки. При его выполнении можно заметить, что оператор *b:=-10;* необходимо внести в тело внешнего цикла, так как закон изменения переменной *b* не соответствует условию. После всех исправлений программа будет иметь вид, приведенный на рис.10.



Рис.7. Пошаговое выполнение программы (локализация ошибки “деление на нуль”)

The screenshot shows a Pascal program named PP1.PAS being executed. The program contains a loop that repeats until variable 'a' is greater than 1000. Inside the loop, it checks if 'b' is zero. If 'b' is zero, it prints an error message. Otherwise, it calculates 'y' as 'a' divided by 'b', increments 'b' by 1, and prints the current values of 'a', 'b', and 'y'. The execution has stopped at line 9 because 'b' is 0, causing a division by zero error. The current state of variables is shown at the bottom: a = 100, b = 0, y = -100.0.

```
File Edit Search Run Compile Debug Tools Options Window Help
\PP1.PAS 1
< 1> Program PP1;
< 2>   Uses Crt;
< 3>   Var a,b:Integer; y:Real;
< 4>   Begin
< 5>     a:=100;
< 6>     b:=-10;
< 7>     Repeat
< 8>       Repeat
< 9>         if b = 0 then Writeln('a=',a:5,' b=',b:5,'Error')
<10>       else
<11>         begin
<12>           y:=a/b;
<13>           b:=b+1;
<14>           Writeln('a=',a:5,' b=',b:5,' y=',y:6:2);
<15>         end;

a: 100
b: 0
y: -100.0
```

Рис.8. Пошаговое выполнение программы (вставлен оператор контроля “деление на ноль”)

The screenshot shows the same Pascal program as in Figure 8, but with several corrections. The loop structure has been changed from a nested 'Repeat' loop to a 'Repeat...Until' loop. The division by zero check is now inside the loop body. The program includes a 'Delay(500);' statement to pause execution. The loop continues until 'b' is greater than 10 and 'a' is greater than 1000. The program ends with 'End.'.

```
File Edit Search Run Compile Debug Tools Options Window Help
\PP1.PAS 1=[↑]
< 1> Program PP1;
< 2>   Uses Crt;
< 3>   Var a,b:Integer; y:Real;
< 4>   Begin
< 5>     a:=100;
< 6>     b:=-10;
< 7>     Repeat
< 8>       Repeat
< 9>         if b = 0 then Writeln('a=',a:5,' b=',b:5,'Error')
<10>       else
<11>         begin
<12>           y:=a/b;
<13>           Writeln('a=',a:5,' b=',b:5,' y=',y:6:2);
<14>         end;
<15>         Delay(500);
<16>         b:=b+1;
<17>       Until b>10;
<18>       a:=a+10;
<19>     Until a>1000;
<20>   End.
```

Рис.9. Исправленный текст программы



```
File Edit Search Run Compile Debug Tools Options Window Help
\PP1.PAS
< 1> Program PP1;
< 2> Uses Crt;
< 3> Var a,b:Integer; y:Real;
< 4> Begin
< 5>   a:=100;
< 6>   Repeat
< 7>     b:=-10;
< 8>     Repeat
< 9>       if b = 0 then Writeln('a=',a:5,' b=',b:5,'Error')
<10>     else
<11>       begin
<12>         y:=a/b;
<13>         Writeln('a=',a:5,' b=',b:5,' y=',y:6:2);
<14>       end;
<15>       Delay(500);
<16>       b:=b+1;
<17>     Until b>10;
<18>     a:=a+10;
<19>   Until a>1000;
<20> End.
```

* 7:22

F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

Рис.10. Окончательный текст программы

3. Варианты индивидуальных заданий по лабораторной работе № 1

Индивидуальные задания по лабораторной работе № 1 объединены в группы. В каждой группе собраны задания одинаковой тематической направленности.

3.1. Первая группа заданий

3.1.1. Варианты индивидуальных заданий

Для заданного преподавателем пункта приведенной ниже таблицы вычислить значение функции $f(x,n)$ для $n = 10; 11; \dots 15$ и значения x , изменяющегося от $x_n = 0.6$ до $x_k = 1.1$ с шагом $\Delta x = 0.25$. Результат вывести на печать в виде:

n	Значение	x	Значение	f	Значение
-----	----------	-----	----------	-----	----------

Варианты индивидуальных заданий

N п/п	$Y = f(x, n)$
1	2
1	$e^x + \sum_{k=1}^n \cos(kx)$
2	$\sqrt{x} + \frac{1}{9} \sum_{k=1}^n e^{\sqrt[k]{x^{k-1}}} + \sqrt{x^k}$
3	$\arctg x - \sum_{k=1}^n \frac{x^{2k-1} + 1/7}{\sqrt{e^{x/k}} + \sqrt[k]{x^{k-1}}}$
4	$\sqrt[3]{e^x + e^{-x}} \sum_{k=1}^n \left(\frac{\sqrt{x}}{k} + \sin \frac{\ln(k+x)}{k} \right)$
5	$\operatorname{tg}(\sin x) + \sum_{k=1}^n \frac{\lg(x^{k-1} + \sqrt{e^{k+x}})}{k+x}$
6	$\frac{\pi}{\sqrt[3]{x} + \frac{3}{x+5}} \sum_{k=1}^n \left(\sin^2 \frac{k-1}{k+1} + e^{\sqrt{x/k}} \right)$
7	$\sqrt{\frac{\pi}{2} + x} \sum_{k=1}^n \frac{e^{\sqrt[3]{k-x}}}{k^2 + x}$
8	$\frac{1}{x^2 + \sqrt{x}} + \sum_{k=1}^n \left(\left(\frac{\sqrt[3]{x}}{k} - e^{-kx} \right) \sin(x+k) \right)$
9	$\sqrt{\sin^3 \frac{x}{n}} \sum_{k=1}^n \frac{1 + \frac{k+1}{n}}{e^{\sqrt[k]{x^{k-1}}} + \ln x}$
10	$ne^x + \sum_{k=1}^n \frac{\cos \sqrt{kx}}{\sqrt[3]{k^2 - \frac{1}{3}}}$

1	2
11	$\sqrt[3]{n + \operatorname{tg} x} \sum_{k=1}^n \frac{x^{2k-1}}{\sqrt{e^{kx} - \frac{1}{k}}}$
12	$\sqrt{n^x + 1} + \sum_{k=1}^n \frac{\sqrt[5]{x} + \cos\left(\frac{kn}{5}\right)}{k - \ln x}$
13	$\sqrt[3]{n\sqrt{x}} - \sum_{k=1}^n \left(\frac{k-1}{k} + e^{ \cos kx } \right)$
14	$\ln nx - 1 - \sum_{k=1}^n \frac{\sqrt{e^{-kx} + x^{k-1}}}{\cos kx + \sin kx}$
15	$n^2 \sqrt[n]{x} \sqrt[3]{x^2} + \sum_{k=1}^n \frac{\ln \sqrt{x^2}}{2 - \frac{1}{k}}$
16	$\sin \frac{\pi n}{x+3} \sum_{k=1}^n \frac{\sqrt[k]{x^{k+1}} + \sqrt[k]{e^{k-2/3}}}{1 + \ln x}$
17	$\sqrt[3]{1 + \cos^2 \frac{\pi x}{n}} \sum_{k=1}^n \frac{e^{2/3-k}}{\sqrt{x^k + x^{2k-1}}}$
18	$\sqrt{e^{x/n}} \sum_{k=1}^n \frac{\operatorname{arctg} \left(x - \frac{k^2}{k^2 + 1} \right)}{\sqrt[k]{e^{k+x}}}$
19	$\sqrt{x + \frac{n-1}{n}} + \sum_{k=1}^n \frac{\sqrt[5]{e^{kx-3}}}{\ln kx + \frac{3}{5 + \ln kx}}$
20	$\sqrt{e^{x/n}} + \sqrt{x} - \sum_{k=1}^n \frac{e^{\sqrt[3]{k-x}}}{k^2 + x}$

1	2
21	$\left(\frac{n}{17} + \sqrt{x}\right) \sum_{k=1}^n \frac{\sqrt[k]{e^{1,2k} - \frac{k-1}{k}}}{n + \ln \sqrt{kx}}$
22	$\frac{1}{3} \sqrt{2^x} \sum_{k=1}^n \left(\ln^2 x+k \cos \frac{k^2+x}{n} \right)$
23	$\sqrt{n^x + x + 1} - \sum_{k=1}^n \frac{\lg(x^{k-1} + \sqrt{e^{k+x}})}{k+x}$
24	$n \sqrt[3]{x^2} + \frac{n^x}{5} + \sum_{k=1}^n \frac{\ln \sqrt{x^2}}{2 - \frac{1}{k}}$
25	$\sin \frac{\pi n}{x} - \sum_{k=1}^n \frac{x^{2k-1} + 1/7}{\sqrt{e^{x/k}} + \sqrt[k]{x^{k-1}}}$
26	$ne^x + \sqrt{n^x} + \sum_{k=1}^n \frac{\cos \sqrt{kx}}{\sqrt[3]{k^2 - \frac{1}{3}}}$
27	$\sqrt[3]{1 + \sin^2 \frac{\pi x}{n}} \sum_{k=1}^n \frac{e^{2/3-k}}{\sqrt{x^k + x^{2k-1}}}$
28	$\arctg x + \sqrt{n \sqrt[3]{x}} - \sum_{k=1}^n \frac{\sqrt{e^{-kx} + x^{k-1}}}{\cos kx + \sin kx}$
29	$\frac{1}{5} \sqrt{2^{nx}} \sum_{k=1}^n \left(\ln^2 x+k \cos \frac{k^2+x}{n} \right)$
30	$\cos \frac{\pi x}{n} \sum_{k=1}^n \frac{\sqrt[k]{x^{k+1}} + \sqrt[k]{e^{k-2/3}}}{1 + \ln x}$
31	$\sqrt{e^{2x/n}} \sum_{k=1}^n \left(\frac{\sqrt{x}}{k} + \sin \frac{\ln(k+x)}{k} \right)$

1	2
32	$\sqrt{x^{2/n}} \sum_{k=1}^n \left(\frac{\sqrt{x + \sin(nx)}}{k - \operatorname{tg}(kx)} + \cos \frac{\ln(kx)}{k} \right)$
33	$e^{x + \sin(kx)} \sum_{k=1}^n \frac{\sqrt{e^{-kx} + x^{k-1}}}{(\cos kx + \sin kx) \ln(nx)}$
34	$\left(\frac{e^{nx}}{21} + \sqrt{nx} \right) \sum_{k=1}^n \frac{\sqrt[k]{e^{1.2k} - \frac{k-1}{k+1}}}{\sqrt{k} \sqrt[3]{k} + \ln \sqrt{kx}}$
35	$\frac{1}{9} \sqrt{2^{x+n}} \sum_{k=1}^n \left(\ln^2 x+k \sin \frac{k^2 + \sqrt[4]{x}}{k} \right)$
36	$10 - e^{nx} \sum_{k=1}^n \frac{\sqrt{e^{kx} + \sin x^{k-1}}}{\sin kx \cos(nx)}$
37	$\frac{1}{45} \sqrt{3^{x+n}} \sum_{k=1}^n \left(\log^2 x+k \cos \frac{k + \sqrt[5]{\sin x}}{kx} \right)$
38	$\sin \frac{x}{n} \sum_{k=1}^n \frac{\sqrt[n]{\sqrt{x^{k+1}} + \sqrt[k]{e^{xk}}}}{\log x}$
39	$xn \sqrt{e^x} \sum_{k=1}^n \frac{x^{3k} + 1/56}{\sqrt{e^{x/k} + \sqrt[k-1]{x^3}}}$
40	$\cos \frac{nx}{n} - \sum_{k=1}^n \frac{2x^{4k-1}}{\sqrt{e^{3x/k} + \sqrt[k+1]{x^{2k}}}}$
41	$\operatorname{tg} x + \sqrt[3]{\sqrt{nx}} - \sum_{k=1}^n \frac{\sqrt{e^{kx} - kx^{k+1}}}{n \sin^2 kx}$
42	$n \sqrt[3]{x^2} - \frac{xn^x}{5^n} - \sum_{k=1}^n \frac{\ln \sqrt{x^2}}{3 + \sqrt[n]{kx}}$

3.1.2. Пример выполнения индивидуального задания

Существует несколько способов выполнения задания из таблицы.

Первый способ. Рассчитывается функция для всех сочетаний значений аргументов. Для этого должны быть организованы три цикла – по переменным x , n и k . Степень вложенности циклов равна 3. Внешний цикл организуется по переменной x , в него вложен цикл по переменной n , внутренний цикл организован по переменной k .

Предложенный способ отображает схема вложенности циклов, представленная на рис.11. Для каждой переменной после знака « $=$ » указано ее начальное значение, в скобках шаг, на который эта переменная будет увеличиваться, и далее конечное значение переменной.

$$\begin{array}{l} x = 0.6(0.25)1.1 \\ \left[\begin{array}{l} n = 10(1)15 \\ \left[\begin{array}{l} k = 1(1)n \end{array} \right] \end{array} \right] \end{array}$$

Рис.11. Схема вложенности циклов для первого способа

Однако у данной схемы есть один существенный недостаток -- дублирование при счете: для фиксированного значения x при изменении n от 10 и далее до 15 будет повторно просчитываться сумма первых десяти слагаемых ряда. Всего же в результате реализации схемы (см. рис.11) значение выражения под знаком Σ будет просчитано:

$$\underbrace{3}_{\text{вхождения по } x} * \underbrace{(10 + 11 + 12 + 13 + 14 + 15)}_{\text{вхождения по } k \text{ и } n} = 225 \text{ раз.}$$

Схемы алгоритма решения 1-го варианта индивидуального задания (см. таблицу), основанные на применении первого способа его выполнения, представлены на рис.12 и 13. На рис.12 используются циклы с предусловием, на рис.13 – с постусловием.

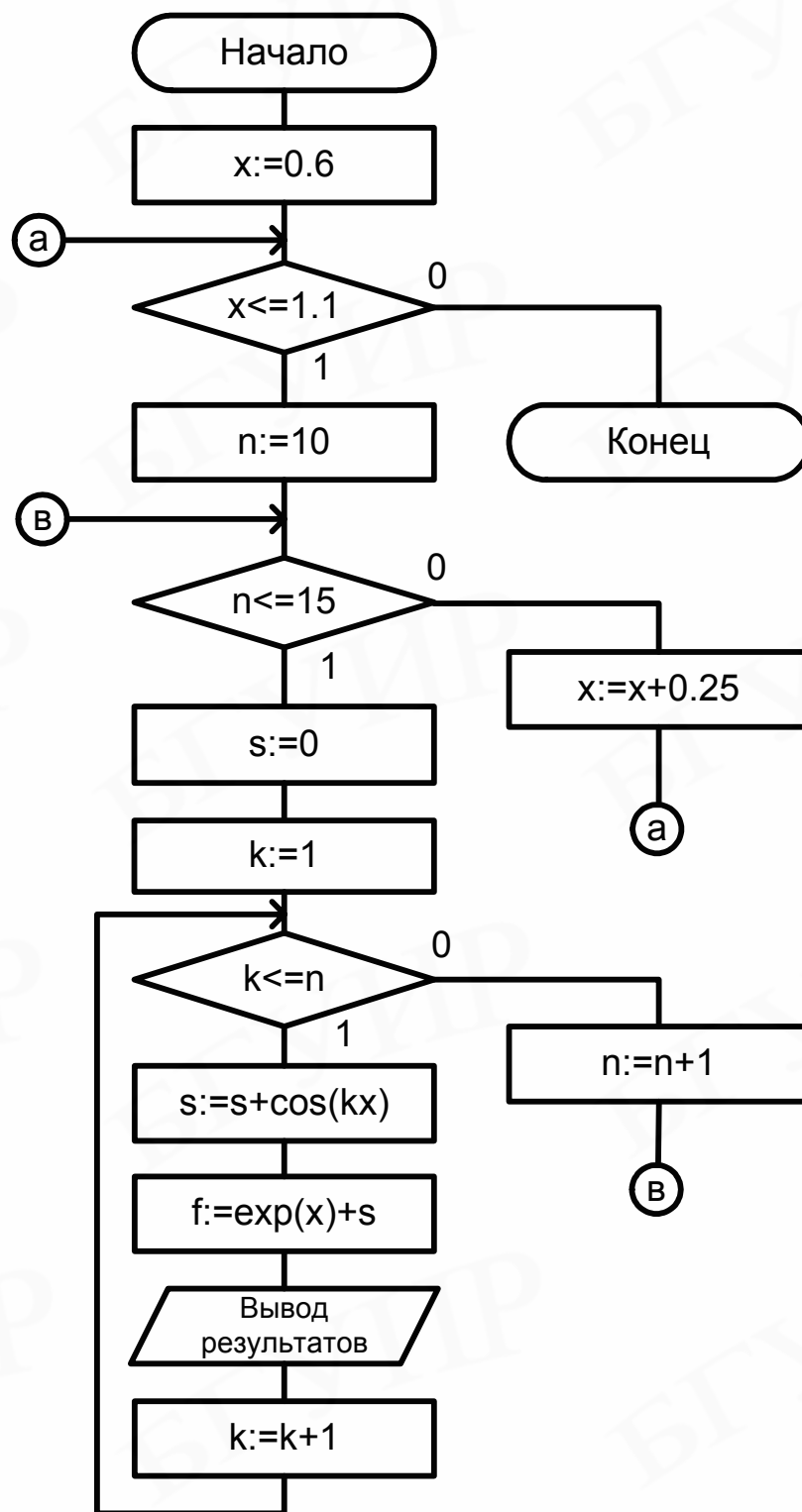


Рис.12. Схема алгоритма в соответствии с ГОСТ 19.701-90
(представление циклов с помощью символа “Решение”;
организация циклов с предусловием)

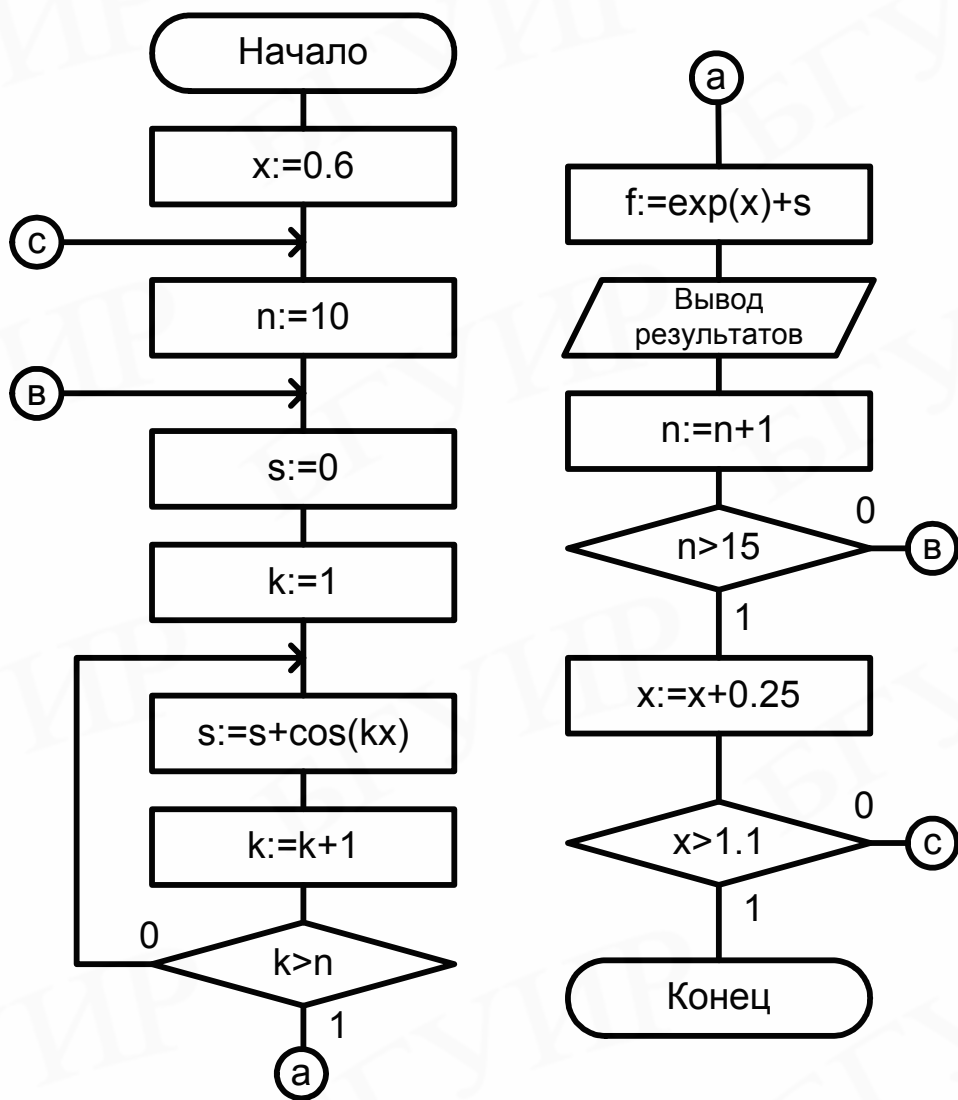


Рис.13. Схема алгоритма в соответствии с ГОСТ 19.701-90
(представление циклов с помощью символа “Решение”;
организация циклов с постусловием)

Текст программы на языке Pascal, реализованный в соответствии со схемой алгоритма, представленной на рис.12, имеет вид:

```

Program Example1;
Var
  k: Integer;
  x, s, f: Real;
Begin
  x:= 0.6;
  While x<= 1.1 do
    Begin
      For n:= 10 to 15 do
        Begin
          s:=0;

```

```

        For k:=1 to n do
            s:= s + Cos( k*x);
        f:= Exp(x) + s;
        Writeln ('N =', n: 4, ' X=', x: 6: 2, ' F =', f: 9: 5)
        End;
        x:= x + 0.25
    End
End.

```

Текст программы на языке Pascal, реализованный в соответствии со схемой алгоритма, приведенной на рис.13, выглядит следующим образом:

```

Program Example2;
Var
    k: Integer;
    x,s,f: Real;
Begin
    x:= 0.6;
    Repeat
        n:= 10;
        Repeat
            s:=0; k:=1;
            Repeat
                s:= s + Cos(k*x);
                k:= k + 1
            Until k >n;
            f:= Exp(x) + s;
            Writeln ('N =', n:4, ' X=', x:6:2, ' F =', f:9:4);
            n:=n+1
        Until n>15;
        x:= x + 0.25
    Until x > 1.1
End.

```

Второй способ. Для организации вычислений, как и в предыдущем случае, используется три цикла, однако степень вложенности циклов равна двум. Внешний цикл организован по переменной x , два вложенных цикла по переменной k располагаются последовательно друг за другом. В первом цикле по переменной k реализуются вычисления, общие для всех n при фиксированном значении x . Во втором цикле к текущему значению результата f прибавляется очередное слагаемое и полученное значение выводится на печать.

Схема вложенности циклов, соответствующая данному способу, представлена на рис.14.

$$x = 0.6(0.25)1.1$$

$$\left\{ \begin{array}{l} k = 1(1)9 \\ k = 10(1)15 \end{array} \right.$$

Рис. 14. Схема вложенности циклов для второго способа

В этом случае значение выражения под знаком Σ будет считаться

$$\underbrace{3}_{\text{вхождения по } x} * \underbrace{(9 + 6)}_{\text{вхождения по } k} = 45 \text{ раз.}$$

Схемы алгоритма решения 1-го варианта индивидуального задания (см. таблицу), основанные на применении второго способа его выполнения, приведены на рис.15 и 16.

Текст программы на языке Pascal, написанный в соответствии со схемой алгоритма, представленной на рис.15, имеет вид:

```

Program Example3;
Var
  k: Integer;
  x, s, f: Real;
Begin
  x:= 0.6;
  While x<=1.1 do
  Begin
    s:= 0;
    For k:=1 to 9 do
      s:= s + Cos (k*x);
    For k:= 10 to 15 do
    Begin
      s:= s + Cos (k*x);
      f:= Exp(x) + s;
      Writeln ('N =', k:4, ' X=', x:6:2, ' F =', f:9:4)
    End;
    x:= x + 0.25
  End
End.

```

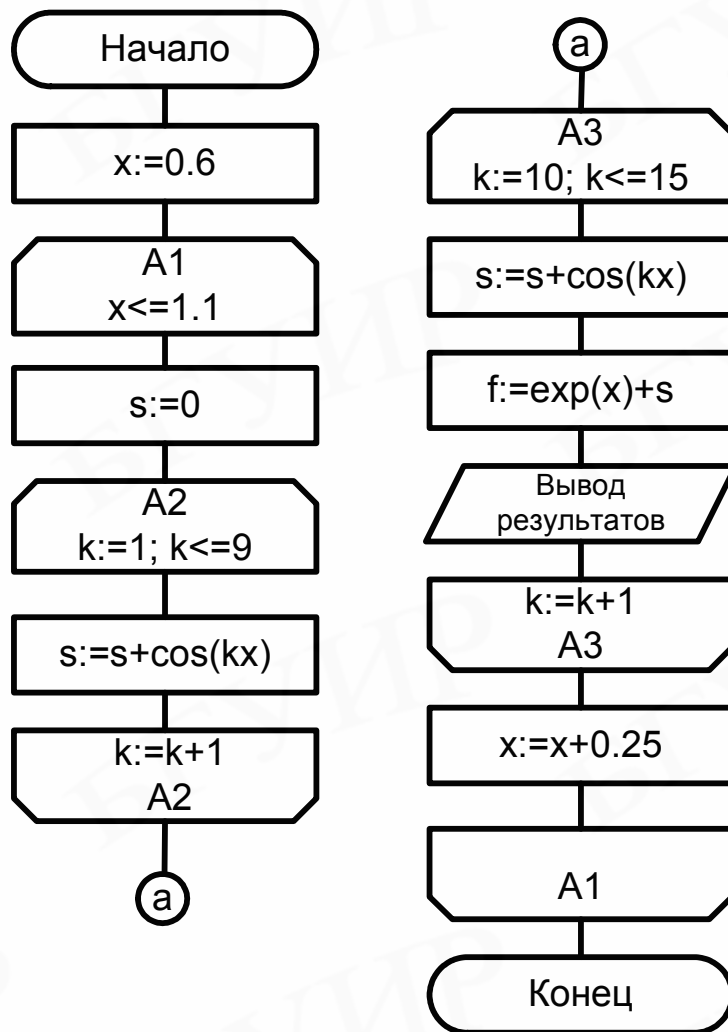


Рис.15. Схема алгоритма в соответствии с ГОСТ 19.701-90
(представление циклов с помощью символа “Граница цикла”;
использование циклов с предусловием)

Текст программы, написанной на языке Pascal в соответствии со схемой алгоритма, представленной на рис.16, имеет вид:

```

Program Example4;
Var
  k: Integer;
  x, s, f: Real;
Begin
  x:= 0.6;
  Repeat
    k:= 1; s := 0;
    Repeat
      s:= s + Cos (k*x);
      k:= k + 1
    Until k >9 ;
  
```

```

Repeat
  s:= s + Cos(k*x);
  f:= Exp(x) + s;
  Writeln ('N =', k:4, ' X=', x:6:2, ' F =', f:9:4);
  k:= k + 1
Until k > 15;
x:= x + 0.25
Until x > 1.1
End.

```

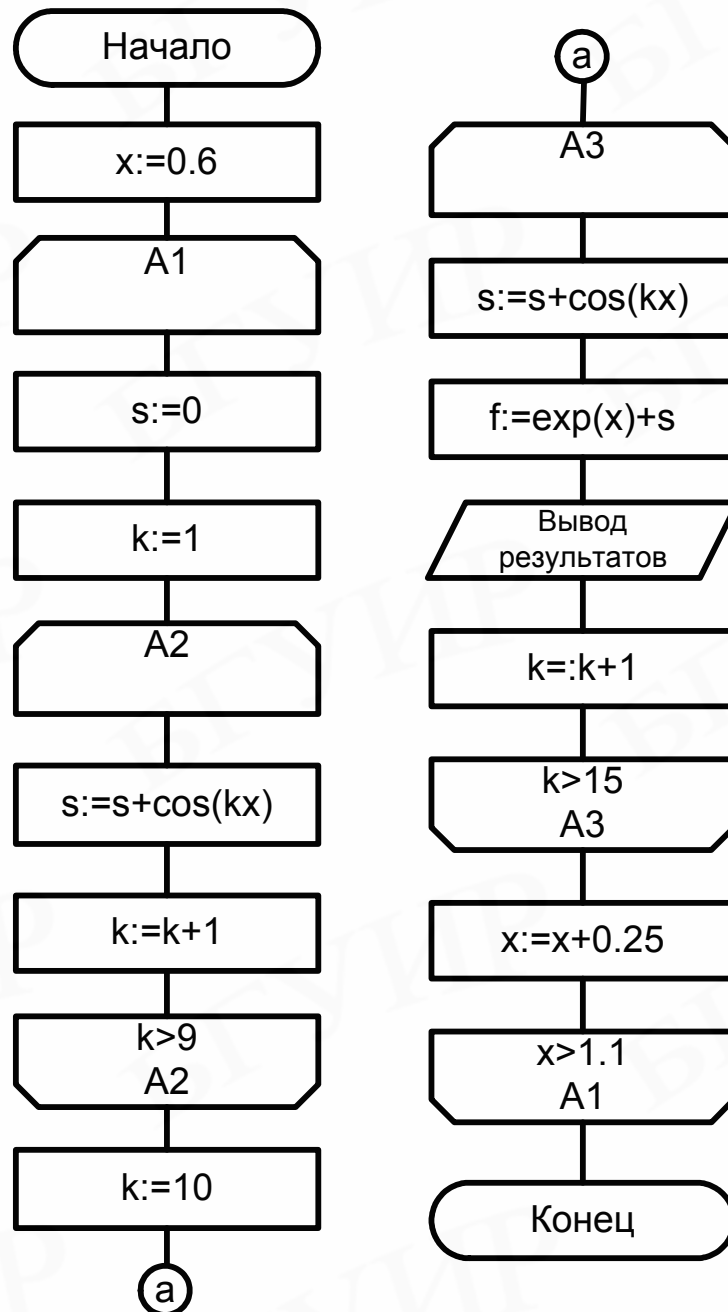


Рис.16. Схема алгоритма в соответствии с ГОСТ 19.701-90
(представление циклов с помощью символа “Граница цикла”;
использование циклов с постусловием)

Третий способ реализации описываемого алгоритма по своей эффективности приближается к предыдущему. Схема вложенности циклов при использовании данного способа представлена на рис.17.

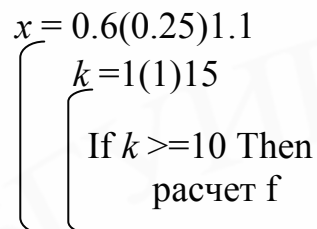


Рис.17. Схема вложенности циклов для третьего способа

При этом способе число циклов сокращается до двух, однако вложенный цикл имеет внутри себя ветвление. Внешний цикл организован по переменной x , вложенный - по переменной k , причем k принимает все значения, вплоть до максимального значения переменной n , равного 15. Если $k = 10$ и более, необходимо вычислить значение функции f и вывести полученные результаты.

В данном случае значение выражения под знаком Σ будет считаться

$$3 * 15 = 45 \text{ раз.}$$

вхождения по x
вхождения по k

На рис.18 – 21 приведены несколько видов схем алгоритма для третьего способа решения задачи первого варианта индивидуального задания (см. таблицу).

Текст программы на языке Pascal, написанный в соответствии с алгоритмами, приведенными на рис.18 и 20, имеет вид:

```

Program Example5;
Var
  k: Integer;
  x,s,f: Real;
Begin
  x:= 0.6;
  While x<=1.1 do
  Begin
    s:= 0;
    For k:=1 to 15 do
    Begin

```

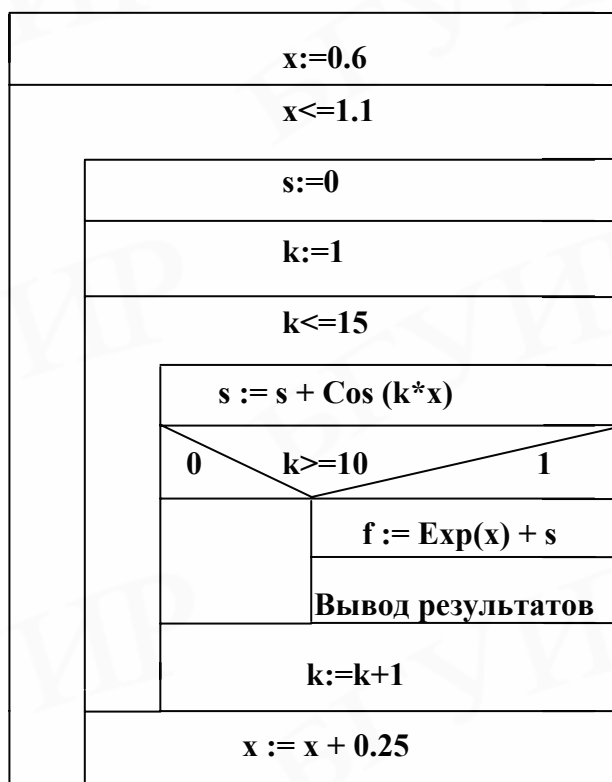


Рис.18. Схема Насси - Шнейдермана (использование циклов с предусловием)

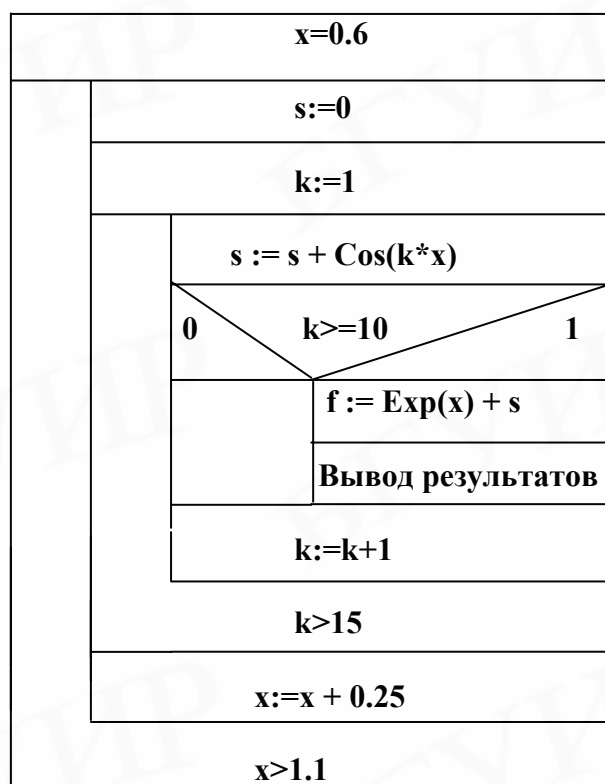


Рис.19. Схема Насси - Шнейдермана (использование циклов с постусловием)

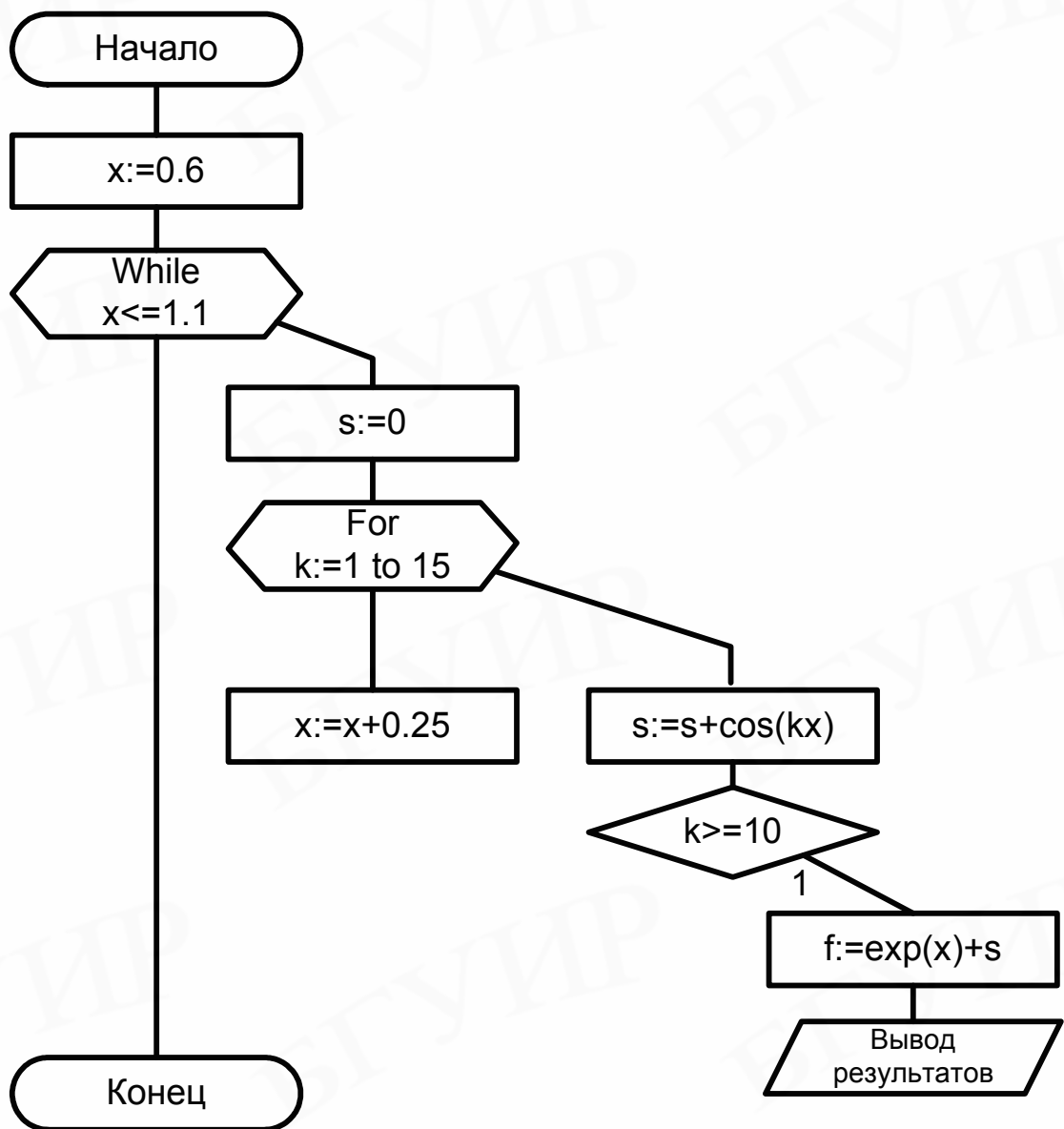


Рис.20. Схема алгоритма, представленная по методу Дамке (использование циклов с предусловием)

```

s:= s + Cos (k*x);
If k>=10 Then
  Begin
    f:= Exp(x) + s;
    Writeln ('N =', k:4, ' X=', x:6:2, ' F =', f:9:4)
  End
End;
x:= x + 0.25
End
End.

```

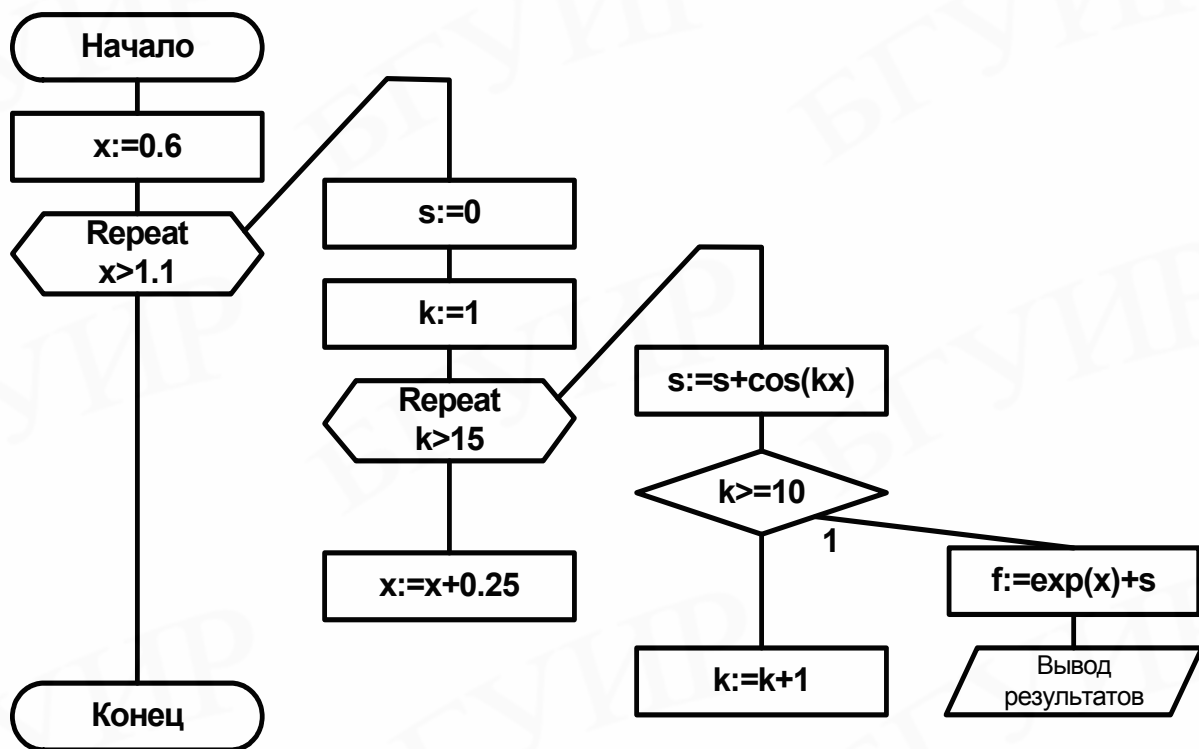


Рис.21. Схема алгоритма, представленная по методу Дамке
(использование циклов с постусловием)

Текст программы на языке Pascal, написанный в соответствии с алгоритмами, приведенными на рис.19 и 21, имеет вид:

```

Program Example6;
Var
  k: Integer;
  x, s, f: Real;
Begin
  x:= 0.6;
  Repeat
    s:= 0; k:=1;
    Repeat
      s:= s + Cos(k*x);
      If k>=10 then
        Begin
          f:= Exp(x) + s;
          Writeln ('N =', k:4, ' X=', x:6:2, ' F =', f:9:4)
        End;
      k:=k+1
    Until k>15;
    x:= x + 0.25
  Until x>1.1
End.

```

3.2. Вторая группа заданий

1. Найти натуральное число от 1 до 10000 с максимальной суммой делителей.

2. Два натуральных числа называют дружественными, если каждое из них равно сумме всех делителей другого, кроме самого этого числа. Найти все пары дружественных чисел, лежащих в диапазоне от 200 до 300.

3. Даны целые числа P и Q . Получить все делители числа Q , взаимно простые с P .

4. Дано натуральное число N . Найти все меньшие N числа Мерсена. (Простое число называется числом Мерсена, если оно может быть представлено в виде $2^p - 1$, где p – тоже простое число).

5. Даны натуральные числа N и M . Получить все меньшие N натуральные числа, квадрат суммы цифр которых равен M .

6. Дано натуральное число N . Среди чисел от 1 до N найти все такие, запись которых совпадает с последними цифрами записи их квадрата (например, $6^2 = 36$, $25^2 = 625$).

7. Натуральное число из n цифр является числом Армстронга, если сумма его цифр, возведенных в n -ю степень, равна самому числу. (Например, $153 = 1^3 + 5^3 + 3^3$). Получить все числа Армстронга, состоящие из двух, трех и четырех цифр.

8. Известно, что любое натуральное число можно представить в виде суммы не более чем четырех квадратов натуральных чисел (иначе в виде суммы четырех квадратов не отрицательных целых чисел). Дано натуральное число N , указать такие неотрицательные целые x, y, z, t , что $n = x^2 + y^2 + z^2 + t^2$.

9. Даны взаимно простые натуральные числа p и q ($p < q$). Найти периодическую и непериодическую части (две последовательности однозначных неотрицательных чисел) десятичной дроби, равной p/q .

10. Записать произвольное (необязательно целое) число в обратном порядке.

11. Найти числа, для которых перестановка последней цифры в начало увеличивает его в N раз.

12. Найти два числа, таких, у которых цифры первого являются перестановкой цифр второго, и, если из первого вычесть второе, то получится сумма цифр одного из них.

13. Найти четырехзначные числа, такие, чтобы квадратный корень из них был равен числу, образованному первыми двумя цифрами в сумме с квадратным корнем из числа, образованного последними его цифрами.

14. Найти натуральные числа, такие, чтобы сумма их цифр, а также сумма цифр следующего за ним числа делилась на 7.

15. Найти все такие M -значные числа ($M=2,3,\dots$), которые делятся на каждую из цифр в их записи.

16. Из 3 цифр составили все возможные числа, сложили их и получили некое M . Определить эти цифры.

17. Составить программу нахождения наименьшего общего кратного (двух и более чисел).

18. Составить программу нахождения наибольшего общего делителя (двух и более чисел).

19. Найти все меньшие 1000 натуральных числа, которые при возведении в квадрат дают палиндром (перевертыш). (Палиндром – число, в записи которого прямой и обратный порядок цифр одинаков).

20. Найти все меньшие 1000 числа-палиндромы, которые при возведении в квадрат также дают палиндром (определение палиндрома см. в задании №19).

21. Получить все меньшие 10^6 натуральных числа, которые являются палиндромами как в десятичной, так и в двоичной системах счисления (определение палиндрома см. в задании №19).

22. Для целого числа M найти и напечатать все простые множители в порядке их возрастания. Одинаковые множители печатать столько раз, сколько они встречаются.

23. Найти 10 пар простых чисел-близнецов (два простых числа называются числами-близнецами, если разница между ними равна 2, например 3 и 5, 11 и 13 и т.д.).

24. Пусть есть некоторое натуральное число. Найти сумму квадратов цифр этого числа, получив новое число, с этим новым числом проделать аналогичную процедуру. После конечного числа повторений этой процедуры получается либо число 1, либо число 4. На промежутке $[1...N]$ найти числа и их количество, которые по завершении вышеописанной процедуры дают результат 1 ($N \leq 30000$).

25. Найти четырехзначное число, являющееся полным квадратом, у которого первые и последние две цифры одинаковы (определение полного квадрата см. в задании №30).

26. Для натуральных чисел из некоторого диапазона от n до m проделать следующую процедуру: переставить цифры в числе, вновь полученное число сложить с исходным. Указанные действия повторять до тех пор, пока не получится палиндром. Напечатать для каждого числа количество действий и свой палиндром (определение палиндрома см. в задании №19).

27. Дано некоторое натуральное число N . Если это не палиндром, то изменить порядок его цифр на обратный и сложить исходное число с полученным. Если сумма – не палиндром, то над ней повторить те же действия и т.д., пока не получится палиндром (определение палиндрома см. в задании №19).

28. Составное число называется числом Смита, если сумма его цифр равна сумме цифр всех чисел, образующихся разложением исходного числа на простые множители. Найти число Смита с номером N .

29. Составное число называется числом Смита, если сумма его цифр равна сумме цифр всех чисел, образующихся разложением исходного числа на простые множители. Число Смита называется супер-числом Смита, если сумма его цифр является числом Смита. Найти супер-число Смита с номером N.

30. Составное число называется числом Смита, если сумма его цифр равна сумме цифр всех чисел, образующихся разложением исходного числа на простые множители. Число Смита называется супер-числом Смита, если сумма его цифр является числом Смита. Найти супер-число Смита с номером N, являющееся полным квадратом. (Число называется полным квадратом, если корень квадратный из него является простым числом).

31. Дано натуральное число m. Найти такое натуральное число n, что двоичная запись числа n получается из двоичной записи числа m изменением порядка цифр на обратный. Число m задано в десятичной системе счисления. Надо получить число n также в десятичной системе счисления (например, для m=6 получается n=3).

32. Рассматриваются числа вида $(n \cdot n - 1)$, где n – натуральное число. Из них выбираются те числа, которые являются произведением k различных простых множителей, $k < 7$. Найти S первых таких чисел, $S < 15$. (Множители не превышают 1000, а числа вида $(n \cdot n - 1) < 2000000000$).

33. Дано натуральное число m. Вставить между некоторыми цифрами 1,2,3,4,5,6,7,8,9, записанными именно в таком порядке, знаки “+” или “-“, чтобы значением получившегося выражения было число m. (Например, если m=122, то подойдет следующая расстановка знаков: $12+34-5-6+78+9$). Если расстановка знаков невозможна, то сообщить об этом.

3.3. Задания для самостоятельной работы

1. Вычислить $f = \frac{y+x}{y-x}$, где

$$y = 2 + \frac{x}{6 + \frac{x}{10 + \frac{x}{14 + \frac{x}{18 + \frac{x}{22}}}}}$$

при изменении x на отрезке $[0;1]$ с шагом 0.01.

2. Вычислить цепную дробь для $X = 0.5(0.05)0.8$:

a)

$$y = \frac{1}{x^2 + \frac{2}{x^2 + \frac{4}{x^2 + \frac{8}{\dots x^2 + \frac{256}{x^2}}}}}$$

б)

$$y = \frac{x}{x^2 + \frac{x^2}{x^3 + \frac{x^3}{x^4 + \frac{x^4}{\dots x^{11} + \frac{x^{11}}{x^{12}}}}}}$$

в)

$$y = \frac{x}{1 - \frac{x-1}{2 - \frac{x-2}{3 - \frac{x-3}{\dots 200 - \frac{x-200}{201}}}}}$$

г)

$$y = \frac{15!(x-15)}{0.25 + \frac{14!(x-14)}{0.5 + \frac{13!(x-13)}{1 + \frac{12!(x-12)}{\dots 2024 + \frac{1!(x-1)}{4048}}}}}$$

д)

$$y = \frac{x^{10} + 1}{1 + \frac{x^9 + 2}{2 + \frac{x^8 + 4}{4 + \frac{x^7 + 8}{\dots}}}}$$

$$256 + \frac{x + 512}{512}$$

е)

$$y = \frac{x^{10}(x - 10)}{\sqrt[10]{x} + \frac{x^9(x - 9)}{\sqrt[9]{x} + \frac{x^8(x - 8)}{\sqrt[8]{x} + \frac{x^7(x - 7)}{\dots}}}}$$

$$\sqrt[3]{x} + \frac{x^2(x - 2)}{\sqrt{x}}$$

ж)

$$y = \frac{x}{1 + x - \frac{x(1 + x)}{2 + 3x - \frac{2^2 x(1 + x)}{3 + 5x - \frac{3^2 x(1 + x)}{\dots}}}}$$

$$10 + 19x - \frac{10^2 x(1 + x)}{11 + 21x}$$

з)

$$y = \frac{x^n}{\sqrt[n]{x} + \frac{x^{n-1}}{\sqrt[n-1]{x} + \frac{x^{n-2}}{\sqrt[n-2]{x} + \frac{x^{n-3}}{\dots}}}}$$

$$\frac{x^2}{\sqrt{x} + 1}$$

и)

$$y = \frac{x-1}{1 + \frac{1(x-1)}{2 + \frac{1(x-1)}{3 + \frac{2^2(x-1)}{4 + \frac{2^2(x-1)}{5 + \frac{3^2(x-1)}{\dots}}}}}}$$

$$12 + \frac{6^2(x-1)}{13}$$

3. Вычислить значения функции (если значение функции не может быть определено для некоторых значений аргументов, напечатать соответствующие сообщения):

$$а) f(a, x) = \begin{cases} \sqrt{2 + \sin^2(x)}, \text{ если } 0 < x \leq 1, \\ \sqrt[4]{a^3 + x^3}, \text{ если } 1 < x < 1.5, \\ \frac{1}{a}, \text{ если } 1.5 \leq x < 2; \\ 1, \text{ иначе} \end{cases}$$

для $a = -1, a = 0, a = 1$. X изменять от 0 до 2 с шагом 0.1.

$$б) f(a, x) = \begin{cases} \frac{\sin(x^2)}{ax^2}, \text{ если } 0 < x < 0.5, \\ \sqrt{\cos(x)}, \text{ если } 0.5 \leq x < 0.8, \\ 1 - ax^3, \text{ если } 0.8 \leq x < 1, \\ a, \text{ иначе} \end{cases}$$

для $a = -1, a = 0, a = 1$. X изменять от -0.5 до 3 с шагом 0.2.

$$в) f(a, x) = \begin{cases} \frac{1-ax}{1+x^2} + \sin(x), \text{ если } 0 < x < 1, \\ a + \sqrt[3]{x} \sqrt{\cos(x)} + 1, \text{ если } 1 \leq x \leq 1.5, \\ \frac{1}{a^2}, \text{ если } 1.5 < x < 2, \\ 1 + a^2, \text{ иначе} \end{cases}$$

для $a = -1, a = 0, a = 1$. X изменять от 0 до 2.5 с шагом 0.1.

4. Используя схему Горнера, вычислить значения полиномов при изменении X на отрезке $[0;1]$ с шагом 0.01:

$$а) y = x^{18} + 2x^{17} + 3x^{16} + 4x^{15} + \dots + 18x + 19;$$

б) $y = x^{18} + 2x^{16} + 4x^{14} + 8x^{12} + \dots + 256x^2 + 512;$

в) $y = a_{18}x^{18} + a_{17}x^{17} + a_{16}x^{16} + a_{15}x^{15} + \dots + a_1x + a_0.$

5. Используя схему Горнера, вычислить выражение

$$y := \sum_{i=1,3,5,\dots}^{19} \frac{x^7 - x^6 + x^5 - x^4 + x^3 - x^2 + x^1}{i^2} + \sum_{j=2,4,6,\dots}^{20} \frac{x^7 - x^6 + x^5 - x^4 + x^3 - x^2 + x^1}{j+2}$$

Значения X сформированы в массив из 20 значений. Начальное значение X равно 0.6. Шаг изменения 0.1.

6. Вычислить Y(x) по следующей формуле:

$$Y(x) = \begin{cases} \Phi(x), & \text{при } |X| < 1/2, \\ \pi/2 - \Phi(\sqrt{1-x^2}), & \text{при } X > 1/2, \\ \Phi(\sqrt{1-x^2} - \pi/2), & \text{при } X > -1/2, \end{cases}$$

где $\Phi(x) = a_0x + a_1x^3 + a_2x^5 + a_3x^7 + a_4x^9 + a_5x^{11}.$

7. Вычислить Г-функцию на отрезке от 1 до 7 с шагом 0.1. На отрезке [2; 3] пользоваться следующей формулой:

$$\Gamma(x) = \sum_{i=0}^7 a_i x^i,$$

$$a_0 = 0.9999; a_1 = 0.4227; a_2 = 0.4117; a_3 = 0.0821;$$

$$a_4 = 0.0721; a_5 = 0.0044; a_6 = 0.0051; a_7 = 0.0016$$

Для $z = x + n > 3$, где $2 < x < 3$, а n – целое положительное число ($n=1\dots 4$), использовать формулу

$$\Gamma(z) = \Gamma(x+n) = (x+n-1)(x+n-2)\dots(x+1)\Gamma(x)$$

8. Значение интеграла $J_n = \frac{1}{e} \int_0^1 x^n e^x dx$ можно вычислить по

рекуррентной формуле $J_n = 1 - nJ_{n-1}$, $n=1,2,\dots$. Начальное приближение интеграла положить равным $J_0 = 1 - \frac{1}{e}$. Подсчитать J_{13} .

9. Вычислить сумму $\sum_{i=0}^{100} x_i$, где $x_i = 0.45x_{i-1}^2 + 0.3x_{i-2} + 0.25x_{i-3};$

$x_0 = x_1 = x_2 = 1$. Напечатать полученную последовательность чисел и их сумму.

10. Отрезок $[0;1]$ разбить на 16 равных частей. Определить, сколько значений функции $y = \cos^2(kx)$, где $k=1,2,\dots,1000$, попадает в каждый промежуток.

11. Найти значение аргумента X для функции

$$y = ae^{bx+cx^2},$$

при котором достигается максимум. Значение аргумента X изменяется от -2 до 2 с шагом 0.2 . Функция имеет один максимум. (Массив не использовать).

12. Найти экстремумы функции $Y(x) = 2x^2 + 8x^2 + 3x + 7$, имеющей один максимум и один минимум, если X изменяется от -4 до 4 с шагом 0.1 . (Массив не использовать).

13. Даны натуральное число n и действительное число X . Среди чисел $e^{\cos(x^{2k})} \sin(x^{3k})$ ($k=1, \dots, n$) найти ближайшее к какому-нибудь целому. (Массив не использовать).

14. Найти совершенные числа на отрезке $[M; N]$. (Число называется совершенным, если оно равно сумме своих делителей, включая 1 , но не включая самое себя).

15. Напечатать последовательность простых чисел, не превосходящих заданное целое число M . (Целое число X является простым, если оно не делится нацело на все целые числа $1 < Y \leq \sqrt{x}$). Использовать алгоритм – “решето Эратосфена”. (Сначала вычеркивают все числа, делящиеся на 2 , исключая само число 2 . Затем берут первое из оставшихся чисел, а именно 3 . Вычеркивают все идущие после него числа, делящиеся на 3 . Далее, оставшимся числом будет 5 . Вычеркивают все идущие после него числа, делящиеся на 5 и т.д. Числа, уцелевшие после всех вычеркиваний, - простые).

16. Дано натуральное число N . С помощью алгоритма “решето Эратосфена” (см. предыдущую задачу) найти четверки меньших N простых чисел, принадлежащих одному десятку (например, $11, 13, 17, 19$).

17. Ввести два числа A и B . Проверить, есть ли у них совпадающие цифры.

18. Дано натуральное число n . Выбросить из записи числа цифры 0 и 5 , оставив прежним порядок остальных цифр. Например, из числа 59015509 должно получиться число 919 .

19. Дано натуральное число N . Получить все натуральные числа, меньшие N и взаимно простые с ним.

20. Найти M -значные числа, делящиеся на N , у которых цифры возрастают.

21. Построить таблицу всех разбиений целого числа на сумму трех слагаемых. Перестановка слагаемых нового способа не дает.

22. Напечатать все представления натурального числа N суммой натуральных чисел. Перестановка слагаемого нового способа не дает.

23. Дано натуральное число N . Получить в порядке возрастания N первых натуральных чисел, которые не делятся ни на какие простые числа, кроме $2, 3$ и 5 .

24. Рассмотрим некоторое натуральное число N ($N > 1$). Если оно четно, то разделим его на 2, иначе умножим на 3 и прибавим 1. Если полученное число не равно 1, то действие повторяется, и т.д., пока не получится 1. До настоящего времени неизвестно, завершается ли этот процесс для любого $N > 1$. Даны натуральные числа k, z, m . Проверить, верно ли, что для чисел из диапазона от k до z процесс завершается не позднее, чем после m таких действий.

25. Напечатать все простые несократимые дроби, заключенные между 0 и 1, знаменатели которых не превышают число 7. (Дробь задается двумя натуральными числами – числителем и знаменателем).

26. Найти все такие простые числа, не превосходящие заданного числа n , двоичная запись которых представляет симметричную последовательность нулей и единиц (начинающуюся единицей!).

27. Для заданного натурального N определить наименьшее число S , которое можно представить в виде суммы $a^n + b^n$ по крайней мере двумя различными способами (a, b – натуральные числа; представления, отличающиеся лишь порядком слагаемых, различными не считаются).

28. Получить первые N натуральных чисел, делителями которых являются только числа 2, 3 и 5.

29. Указать то число заданного множества целых чисел, в двоичном представлении которого больше всего единиц.

30. Среди простых чисел, не превосходящих N , найти такое, в двоичной записи которого максимальное число единиц.

31. Перечислить все натуральные числа, не превосходящие заданного N , в двоичном представлении которых номера ненулевых разрядов образуют арифметическую прогрессию.

32. Перечислить все пары “соседних” простых чисел, не превосходящих N , троичные представления которых получаются друг из друга записью цифр в обратном порядке (первая такая пара – это 5 и 7).

33. Найти все пары двухзначных натуральных чисел M, N , таких, что значение произведения $M \cdot N$ не изменится, если поменять местами цифры каждого из сомножителей (такой парой будет, например, 38 и 83).

34. Заданы три натуральных числа A, B и N . Найти все натуральные числа, не превосходящие N , которые можно представить в виде суммы (произвольного числа) слагаемых, каждое из которых – A или B .

35. Ввести натуральные числа m и n . Напечатать период десятичной дроби m/n . Например, для дроби $1/7$ периодом будет (142857), а если дробь конечная, то ее период состоит из цифры 0.

36. Найти последовательность из 50 нулей и единиц, в которой никакой отрезок не повторяется три раза подряд. Например, в искомой последовательности нигде не должны встречаться такие отрезки, как 000, или 101010, или 101101101.

ЛИТЕРАТУРА

1. ГОСТ 19.701-90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.
2. ГОСТ 19781-90. Обеспечение систем обработки информации программное. Термины и определения.
3. Бахтизин В.В., Глухова Л.А. и др. Методические указания по вычислительной практике и самостоятельной работе по курсам "Программирование" и "Конструирование программ и языки программирования" для студентов специальности "Вычислительные машины, комплексы, системы и сети", "Программное обеспечение ЭВМ и автоматизированных систем" и слушателей спецфакультета переподготовки по направлению "Микропроцессорные системы". Ч. 1-4. – Мн.: МРТИ, 1989-1992.
4. Бахтизин В.В., Глухова Л.А. Лабораторный практикум по курсам "Конструирование программ и языки программирования" и "Программирование" для студентов специальностей "Программное обеспечение ЭВМ и автоматизированных систем", "Вычислительные машины, комплексы, системы и сети". Ч.1: Конструирование программ с использованием процедур. – Мн.: МРТИ, 1993.
5. Бахтизин В.В., Глухова Л.А. Лабораторный практикум по курсам "Конструирование программ и языки программирования" и "Программирование" для студентов специальностей "Программное обеспечение ЭВМ и автоматизированных систем", "Вычислительные машины, комплексы, системы и сети". Ч.3: Конструирование программ с использованием функций. – Мн.: МРТИ, 1995.
6. Введение в язык Паскаль: Учеб. пособие / В.Г.Абрамов, Н.П.Трифонов, Г.П.Трифорова. – М.: Наука, 1988.
7. Зиглер К. Методы проектирования программных систем. – М.: Мир, 1985.
8. Фаронов В.В. Турбо Паскаль 7.0. Начальный курс: Учеб. пособие. – М.: Нолидж, 1999.
9. Электронные вычислительные машины: Практ. пособие для вузов / Под ред. А.Я.Савельева: В 8 кн. Кн. 3: Алгоритмизация и основы программирования. – М.: Высш. шк., 1987.

Учебное издание

Глухова Лилия Александровна,
Фадеева Елена Павловна,
Фадеева Елена Евгеньевна,
Болтак Светлана Владимировна

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

для студентов специальности 40 01 01
“Программное обеспечение информационных технологий”
дневной формы обучения

В 4-х частях

Часть 1

Редактор Т.А.Лейко
Корректор Е.Н.Батурчик

Подписано в печать 10.05.2004.	Формат 60х84 1/16.	Бумага офсетная.
Гарнитура «Таймс».	Печать ризографическая.	Усл.печ.л. 2,67.
Уч.-изд.л. 2,0.	Тираж 100 экз.	Заказ 2.

Издатель и полиграфическое исполнение:
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Лицензия ЛП N 156 от 30.12.2002.
Лицензия ЛВ N 509 от 03.08.2001.
220013, Минск, П. Бровки, 6