

9



Н.Д. Угринович
И.А. Серёгин
О.А. Полежаева

ИНФОРМАТИКА

Рабочая тетрадь

1



ИЗДАТЕЛЬСТВО

БИНОМ

УЧЕНИ

9 КЛАССА

ШКОЛЫ

Н. Д. Угринович,
И. А. Серёгин, О. А. Полежаева

ИНФОРМАТИКА

9 класс

Рабочая тетрадь

Часть 1



Москва
БИНОМ. Лаборатория знаний

УДК 004.9
ББК 32.97
У27

Угринович Н. Д.
У27 Информатика. 9 класс : рабочая тетрадь : в 2 ч. Ч. 1 /
Н. Д. Угринович, И. А. Серёгин, О. А. Полежаева. —
М. : БИНОМ. Лаборатория знаний, 2017. — 96 с. : ил.
ISBN 978-5-9963-3182-6 (Ч. 1)
ISBN 978-5-9963-3184-0

Рабочая тетрадь входит в состав УМК по информатике для 7–9 классов наряду с тетрадями для 7 и 8 классов, учебниками, методическим пособием для учителя и примерной рабочей программой. Тетрадь выполнена в форме, предусматривающей последовательное выполнение учащимися заданий на компьютере и запись ответов на вопросы, а также оформление отчета. Благодаря такой форме организации занятий педагог может оценить полноту выполнения заданий и сформировать у учащихся культуру оформления практической деятельности проектного и исследовательского характера.

Тетрадь состоит из работ по всем темам учебника для 9 класса с опорой на теоретический материал учебника. Дано подробное описание занятий.

УДК 004.9
ББК 32.97

ISBN 978-5-9963-3182-6 (Ч. 1)
ISBN 978-5-9963-3184-0

© ООО «БИНОМ. Лаборатория знаний», 2017

ВВЕДЕНИЕ

Уважаемые ученики!

Вам предлагается рабочая тетрадь по информатике для 9 класса к учебнику Н. Д. Угриновича «Информатика. 9 класс». Тетрадь состоит из лабораторных работ. Каждая работа включает тему, цель работы, перечисление программного обеспечения, задания для выполнения на компьютере и в тетради.

Выполнение каждой лабораторной работы сопровождается оформлением отчёта. Отчёт расположен в тетради в конце работы, а заполняется по мере выполнения работы.

При выполнении лабораторных работ используются разнообразные электронные образовательные ресурсы из следующих открытых коллекций:

- Федеральный центр информационно-образовательных ресурсов (ФЦИОР): <http://fcior.edu.ru>;
- Единая коллекция цифровых образовательных ресурсов (ЕК ЦОР): <http://school-collection.edu.ru>;
- электронная энциклопедия «Википедия» (Свободная энциклопедия): <http://ru.wikipedia.org/wiki>.

В 9 классе большое внимание уделено обучению алгоритмизации и программированию. Вам предлагается составить и выполнить на компьютере большое количество программ. В учебнике рассматриваются объектно-ориентированные среды. Для данной рабочей тетради авторы предлагают язык программирования Паскаль. В электронном приложении к учебнику, расположенном в авторской мастерской Н. Д. Угриновича, И. Ю. Хлобыстовой, И. А. Серёгина (metodist.Lbz.ru/authors/informatika/1), вы найдёте наборы команд программ — заготовки для выполнения лабораторных работ. Вы можете копировать их в среду программирования. В то же время полезно, если вы будете набирать программы самостоятельно.

По выбору учителя вы будете использовать одну из свободно распространяемых систем программирования (интегрированных сред разработки программ), которые можно скачать из Интернета по ссылкам:

<http://pascalabc.net/> — PascalABC.NET;

<http://soft.hi.ru/programmirovanie/pascal-abc/> — Pascal ABC 3.0.1;

<http://besplatnye-programmy.com/311-izuchenie-programmirovaniya-pascal-abc.html> — Pascal ABC 2.2;

<http://pascal.net.ru/SchoolPakMacOSX> — School Pak для Mac OS X;

<http://files.grate.ru/pascal-abc-dlya-linux.html> — Pascal ABC для Linux;

<http://бесплатные-программы-ру.рф/pascal-abc-skachat-besplatno-na-russkom-yazyke-dlya-windows.html> — Pascal ABC на русском языке для Windows 7/8.

В работе с тетрадью вам помогут навигационные значки:



— выбор одного или нескольких ответов;



— запись короткого ответа;



— запись развёрнутого ответа;



— работа на компьютере;



— поиск информации;



— построение графов и схем;



— заготовка из электронного приложения.

Ряд заданий выделен в блоки с пометкой «Задания для дополнительного выполнения».

Успехов!

Лабораторная работа № 1

ЗНАКОМИМСЯ СО СРЕДОЙ ПРОГРАММИРОВАНИЯ

(К параграфу учебника:

1.1. Алгоритм и его формальное исполнение)

Тема: знакомство со средой программирования Pascal ABC.

Цель работы: научиться ориентироваться в среде программирования Pascal ABC, познакомиться с назначением основных вкладок и пунктов меню.

Программное обеспечение¹: компьютер с установленной операционной системой Windows или Linux, текстовый процессор Microsoft Word; среда программирования Pascal ABC.

1. Алгоритм — это
.....
.....
2. Компьютерная программа — это
.....
3. Язык программирования — это
.....
4. Определите, какие достоинства и недостатки есть у каждого из видов трансляторов.



¹ Нужные программы вам на компьютер установит учитель.

Достоинства интерпретаторов:

Недостатки интерпретаторов:

Достоинства компиляторов:

Недостатки компиляторов:

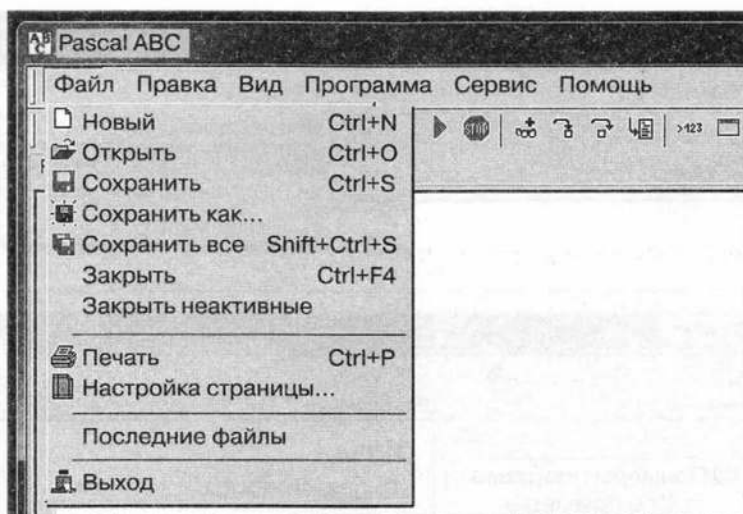


5. Какие трансляторы были созданы раньше — интерпретаторы или компиляторы — и почему? Для ответа можете использовать Интернет и другие источники.





6. Знакомство со средой программирования. Запустите среду программирования (интегрированную среду разработки программ) Pascal ABC. Для этого запустите на выполнение файл PascalABC.exe или загрузите среду посредством ярлыка, если он существует на рабочем столе. Откроется главное окно программы PascalABC. Откройте меню **Файл** и рассмотрите, какие команды оно в себя включает.



Команды:

Новый — создать новый файл для написания новой программы;

Открыть — выбрать из списка и открыть ранее созданный файл;

Сохранить — сохранить файл под старым именем;

Сохранить как — сохранить файл под новым именем;

Печать — вывести содержимое файла на печатающее устройство;

Последние файлы — открыть список нескольких последних созданных файлов;

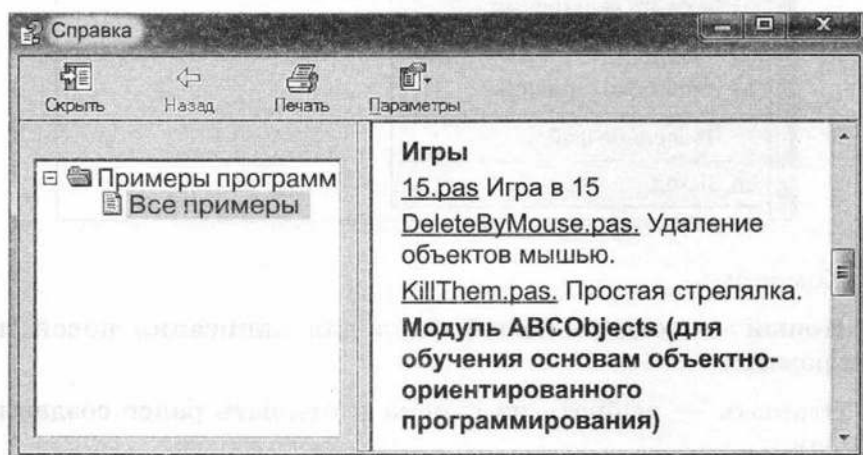
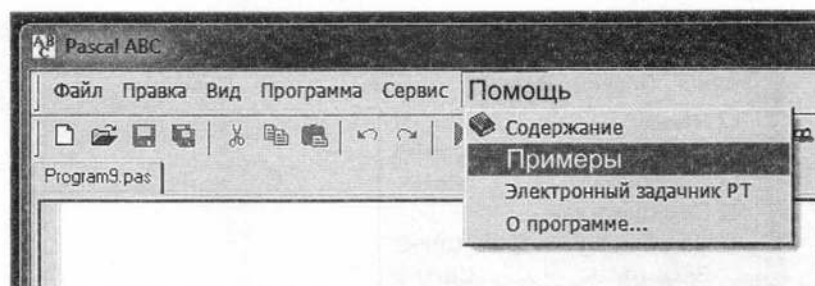
Выход — выйти из среды программирования.

7. Чем различаются команды **Сохранить** и **Сохранить как**?

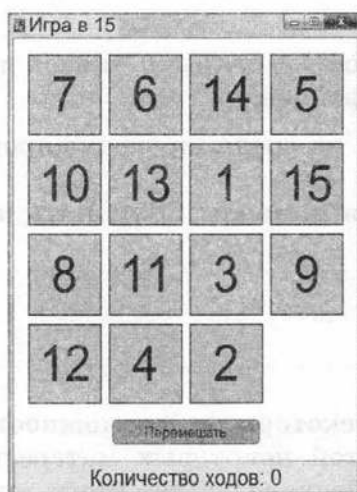


8. Знакомство с некоторыми возможностями языка. Ознакомьтесь с работой некоторых интересных программ, написанных на Паскале. Для этого выполните команду **Помощь** → **Примеры**. Появится окно **Справка** с примерами программ.





Пример 1. Для первого знакомства с Паскалем выберите широко известную **Игру в 15**. В этой игре нужно случайно расположенные фишки расставить в порядке возрастания их номеров.



Запустите программу, нажав кнопку **Выполнить** (зелёный треугольник) или клавишу F9.

Сделайте и сохраните скриншот исходного состояния игры.

Передвигая фишки, постарайтесь решить задачу за наименьшее число ходов.

Сделайте и сохраните скриншот игры после установки первых двух рядов.

Сделайте и сохраните скриншот итогового решения игры.

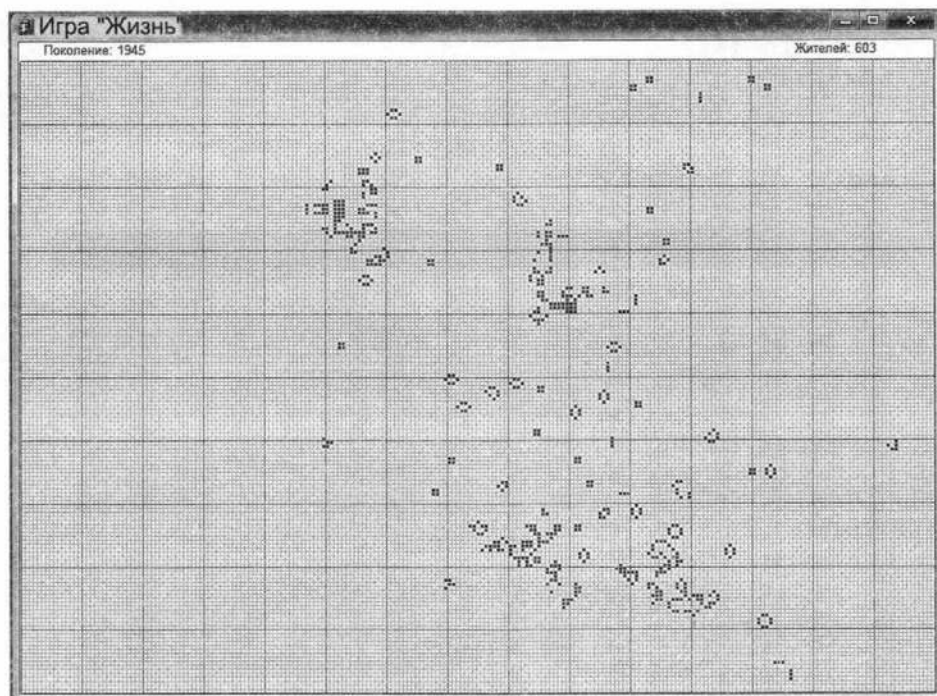
Для дополнительного выполнения

Пример 2. Выберите в окне **Справка** раздел **Модуль GraphABC**.

Выберите интересную моделирующую программу — Игру «Жизнь» («Эволюция») — откройте файл **Life.pas**.

Используя Интернет и другие источники, познакомьтесь с правилами игры Конвея «Жизнь» — законами развития колонии организмов.

Запустите программу и следите за эволюцией колонии организмов.



Сделайте и сохраните скриншот в середине или в конце работы программы. Закройте программу.



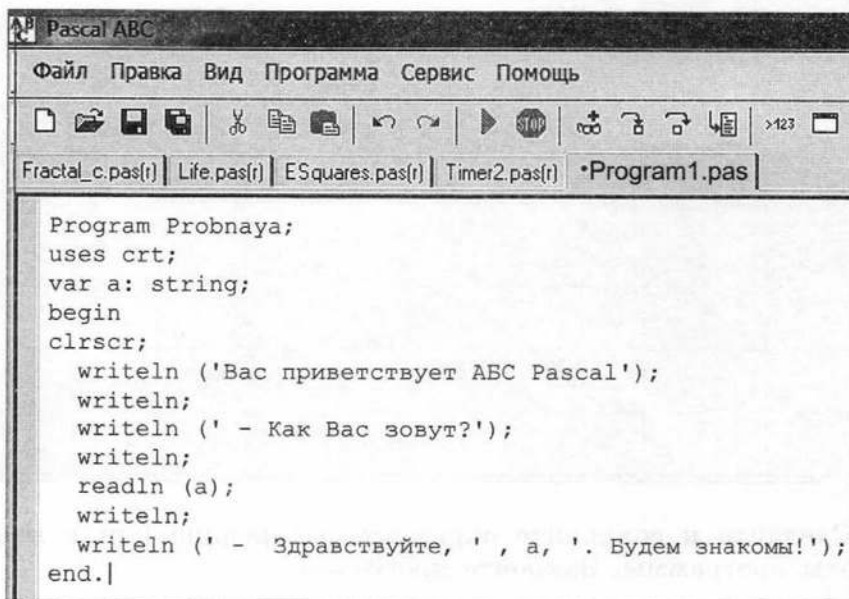
- 9. Первая программа.** Теперь вы напишете свою первую программу. Вначале вы увидите пустой документ (проект, программу) под названием Program1.pas. Наберите приведенную ниже программу.

Для упрощения работы можно скопировать текст программы в среду программирования из электронного приложения к рабочей тетради, но полезнее выполнить его набор.

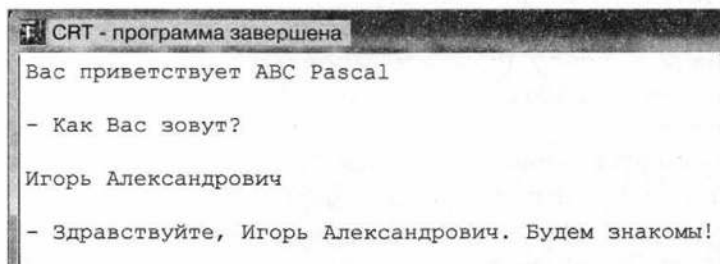
О значении некоторых команд вы сможете догадаться, значение других мы рассмотрим позднее.

www

```
program Probnaya;  
uses crt;  
var a: string;  
begin  
  clrscr;  
  writeln ('Вас приветствует ABC Pascal');  
  writeln;  
  writeln ('- Как Вас зовут?');  
  writeln;  
  readln (a);  
  writeln;  
  writeln ('- Здравствуйте, ', a,  
          ' . Будем знакомы!');  
end.
```



Сделайте и сохраните скриншот набранной программы. Запустите программу. В ответ на заданный вопрос введите свое имя.

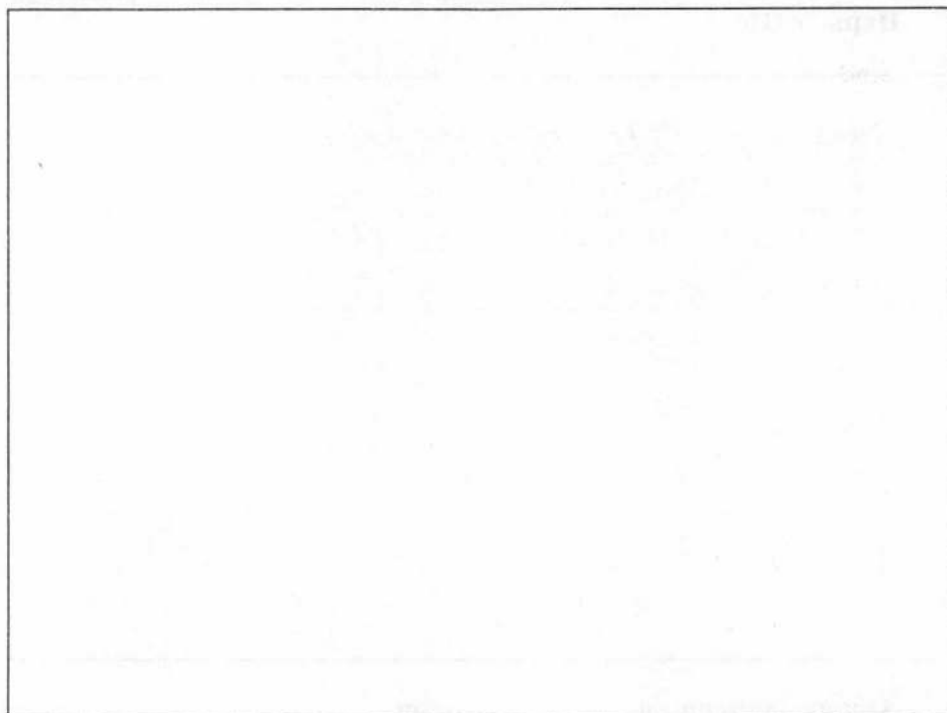


Сделайте и сохраните скриншот результата работы программы.

Сохраните программу в папке, указанной учителем.

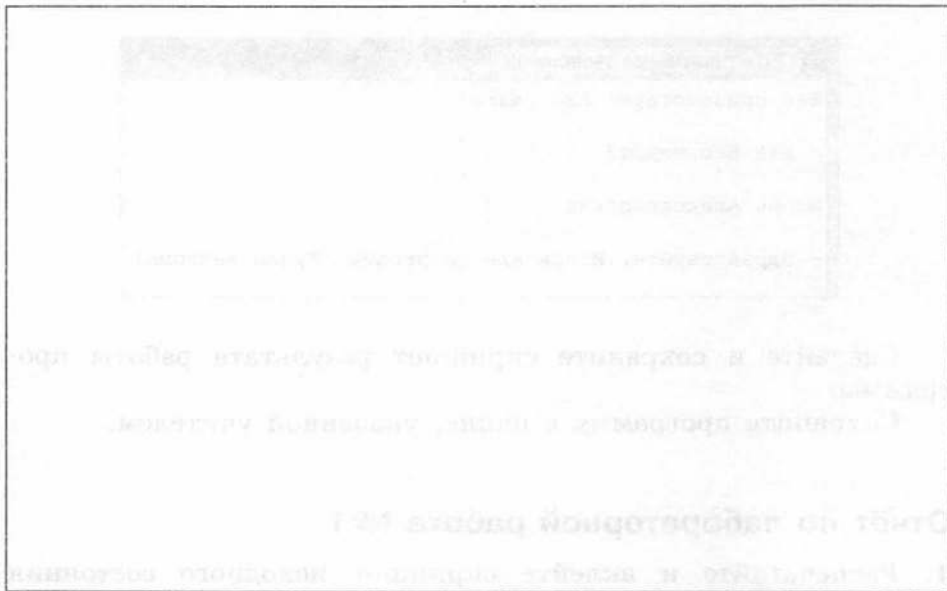
Отчёт по лабораторной работе № 1

1. Распечатайте и вклейте скриншот исходного состояния Игры в 15:

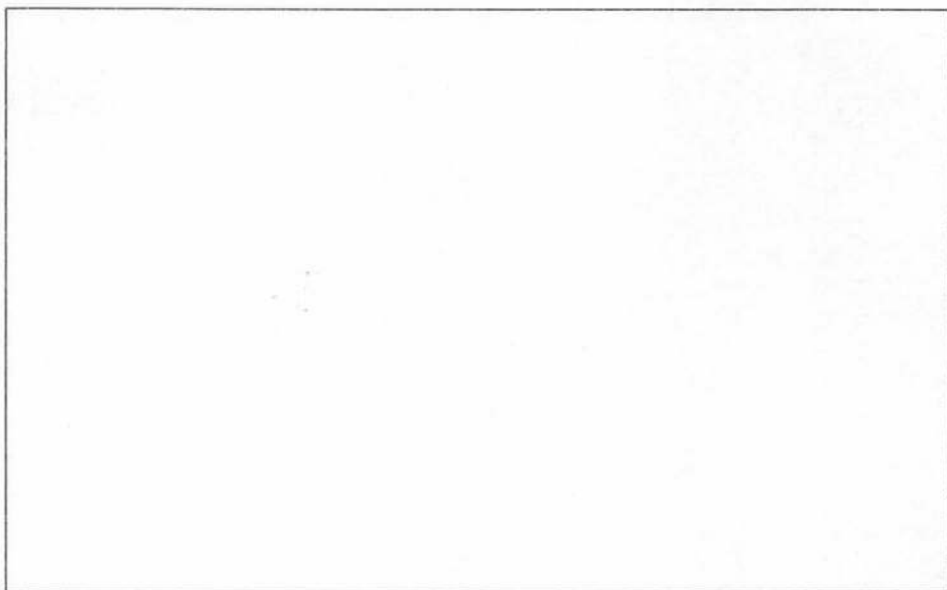




2. Распечатайте и вклейте скриншот **Игры в 15** после установки первых двух рядов:

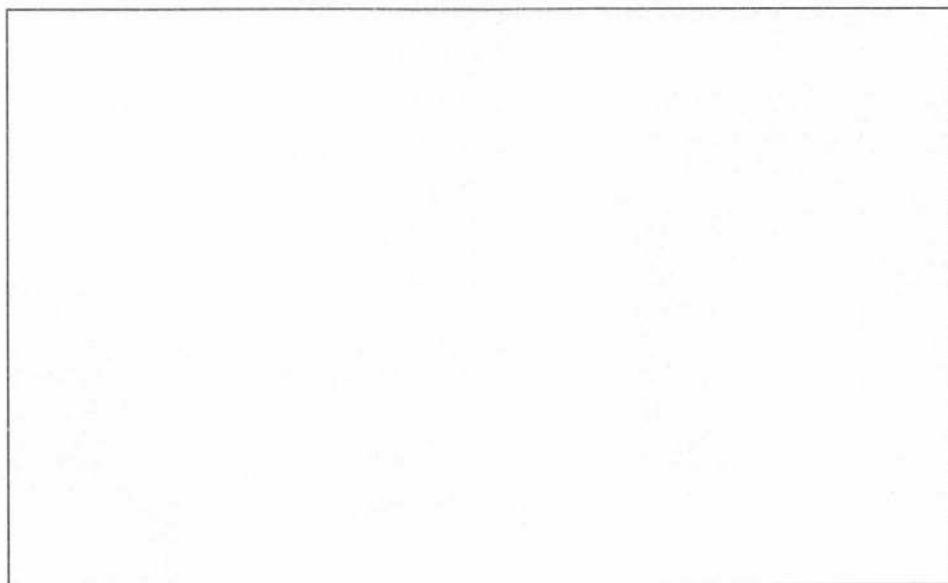


3. Распечатайте и вклейте скриншот **итогового решения** **Игры в 15**:

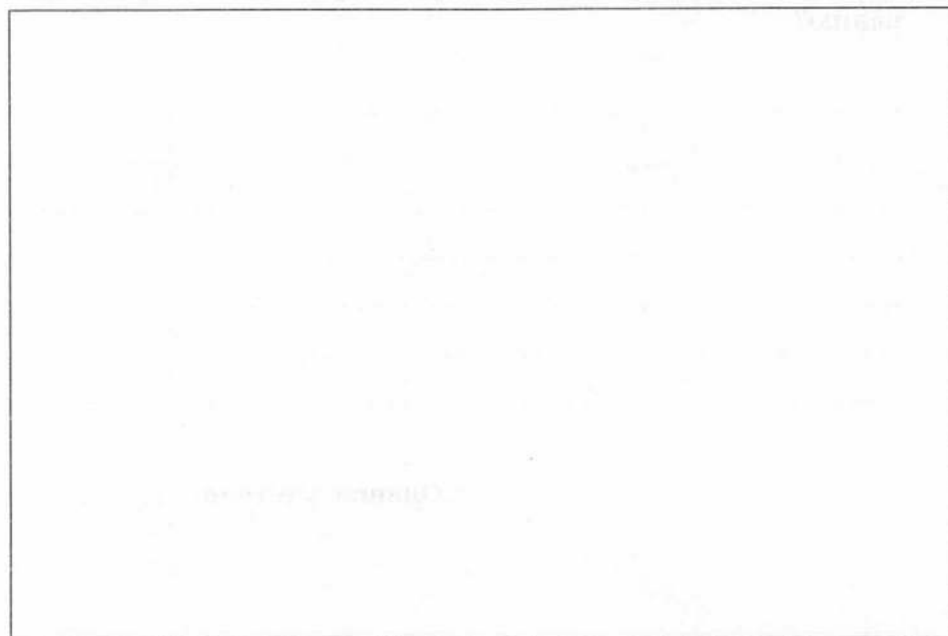


Задача решена за ходов.

4. Распечатайте и вклейте скриншот работы программы «Жизнь» («Эволюция»):



5. Распечатайте и вклейте скриншот вашей набранной программы:





6. Распечатайте и вклейте скриншот результата работы вашей программы:



Выводы

Зачем, по вашему мнению, нужно учиться программированию?

Оценка учителя: -----

Лабораторная работа № 2

УЧИМСЯ ПРОГРАММИРОВАТЬ ЛИНЕЙНЫЕ АЛГОРИТМЫ

(К параграфу учебника: 1.2. Кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования)

Тема: изучение программирования линейных алгоритмов.

Цель работы: научиться составлять и понимать простые программы, используя линейные команды.

Программное обеспечение: компьютер с установленной операционной системой Windows или Linux; программа просмотра ресурсов в Федеральном центре информационно-образовательных ресурсов (ФЦИОР); текстовый процессор Microsoft Word; среда программирования на языке Паскаль.

1. **Знакомство с этапами разработки программы.** На сайте Федерального центра информационно-образовательных ресурсов (ФЦИОР) введите запрос «Этапы разработки программы, её структура. Создание шаблона программы на языке Паскаль».



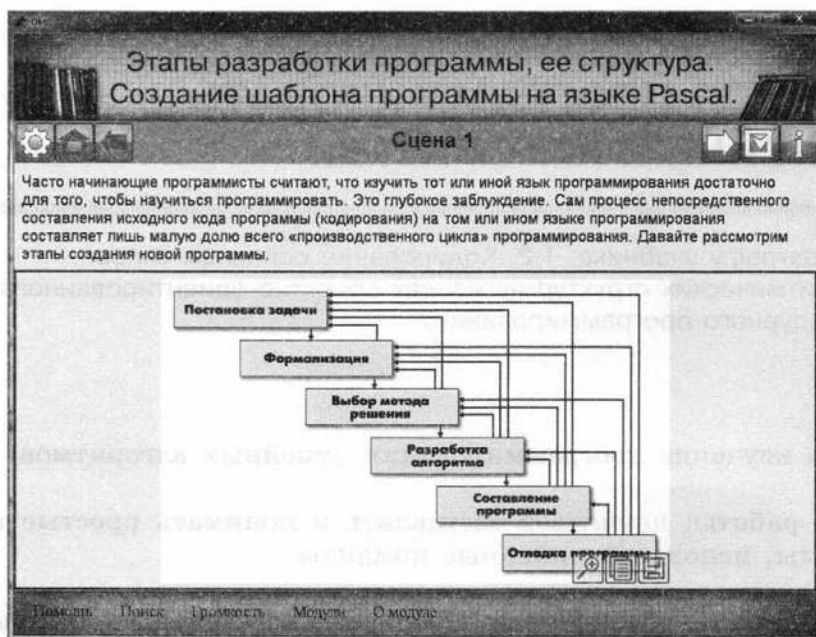
Воспроизведите модуль ресурса:

Этапы разработки программы, её структура. Создание шаблона программы на языке Pascal

Информационный модуль предназначен для получения знаний по этапам разработки программы, её структуре, созданию шаблона программы на языке Pascal.

Тип: Информационный; версия: 1.0.1.8 от 13.04.2010

Ознакомьтесь с информацией из 9 сцен.



В заданиях 2 и 3 и 7 в ответе запишите номер верного варианта.

2. Какой этап является первым при создании новой программы?

- 1) Выбор (или разработка) метода решения
- 2) Разработка алгоритма решения задачи
- 3) Формализация (математическая постановка)
- 4) Составление программы

Ответ:

3. Что используется для имён переменных в Паскале?

- 1) Слова, состоящие из английских букв
- 2) Зарезервированные (служебные) слова
- 3) Слова, состоящие из русских букв
- 4) Слова, состоящие из цифр и латинских букв
- 5) Слова, состоящие из латинских и русских букв

Ответ:

4. Что такое и для чего нужна трассировка программы?

.....

.....

.....

5. Для чего нужно тестирование программы?



6. Что должно учитываться при составлении тестов?



7. Какой знак используется для разделения операторов в Паскале?



1) , 2) . 3) : 4) ; 5) / 6) :=

Ответ:

8. Имеются переменные в операторах:



A:='2'; b:=2; c:=true; d:=123458; e:=25.45678;
f:='25'; g:='pascfl'; h:=0.0000003; i:='i'; j:='false';

Запишите, какие переменные к каким из типов integer, real, char, boolean могут быть отнесены.

integer:

real:

char:

boolean:

9. Укажите номера правильных записей вывода на Паскале.



1) write ln('a=', a); 2) print('a=', a);

3) println(a); 4) writeline(a);

5) WRITE(f); 6) writeln(f).

Ответ:

10. Определите значение переменной a после исполнения алгоритма (из материалов ГИА).



a:=4;

b:=8+2*a;

a:=b/2*a;

Ответ:



11. Определите значение переменной a после исполнения алгоритма (из материалов ГИА).

$a := 16;$
 $b := 12 - a/4;$
 $a := a + b * 3;$

Ответ:



12. Определите значение переменной a после исполнения алгоритма (из материалов ГИА).

$a := 4$
 $b := 10$
 $a := b - a * 2$
 $b := 24 / a * 4$

Ответ:

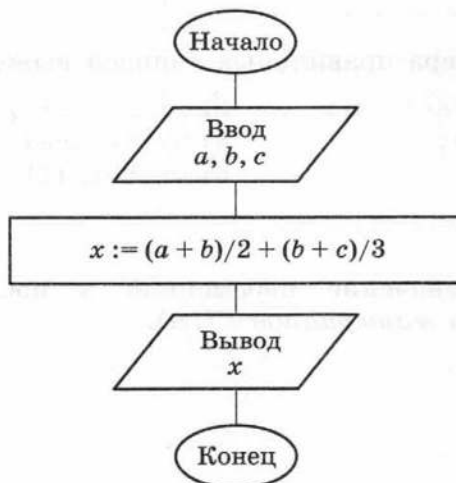
13. Примеры решения задач на Паскале. Ознакомьтесь с примерами решения задач на Паскале.

Пример 1. Вводятся три целых числа a , b , c . Вычислите значение переменной x :

$$x = \frac{a + b}{2} + \frac{b + c}{3}.$$

Решение

Блок-схема:



Программа на Паскале:

```

program raschet_x;
var a, b, c: integer;
    x: real; {так как при делении может получиться
              дробное число, то результат
              должен иметь тип real}

begin
  writeln ('Введите три числа');
  readln (a, b, c);
  x:=(a+b)/2+(b+c)/3;
  writeln ('x = ', x:5:2)
end.

```

Запись $x:5:2$ в операторе вывода означает, что в выводимом дробном числе будет 5 знаков, из них две цифры после запятой.

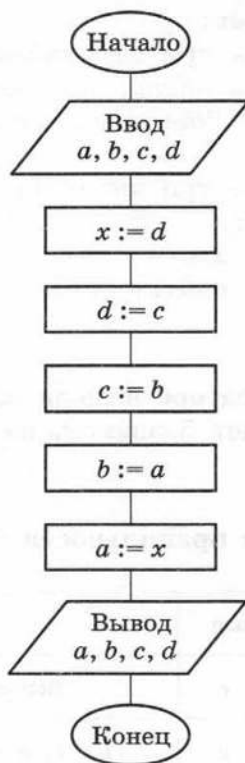
Тесты для проверки правильности программы:

№ теста	Входные данные			Результат	
	<i>a</i>	<i>b</i>	<i>c</i>	Вычисления	<i>x</i>
1	1	1	2	$(1 + 1)/2 + (1 + 2)/3 =$	2,00
2	2	2	2	$(2 + 2)/2 + (2 + 2)/3 =$	3,33
3	0	0	2	$(0 + 0)/2 + (0 + 2)/3 =$	0,67
4	0	0	0	$(0 + 0)/2 + (0 + 0)/3 =$	0,00
5	-1	-1	-2	$(-1 - 1)/2 + (-1 - 2)/3 =$	-2,00

Замечание. Так как исходные данные объявлены как integer, при вводе данных нужно следить, чтобы они не выходили за интервал, определённый для этого типа.

Пример 2. Вводятся четыре переменные *a*, *b*, *c*, *d*. Переставить их в порядке:

→ $a \rightarrow b \rightarrow c \rightarrow d$ →

*Решение***Блок-схема:****Программа на Паскале:**

```
program perestанovka;  
var a, b, c, d, x: integer;  
begin  
  writeln ('Введите четыре числа');  
  readln (a, b, c, d);  
  x:=d;  
  d:=c;  
  c:=b;  
  b:=a;  
  a:=x;  
  writeln (a, ' ', b, ' ', c, ' ', d);  
end.
```

Замечание. Если в операторе вывода не поставить пробелы, то все числа сольются в одно.

Трассировка программы:

№ команды	Команда	Значения переменных				
		<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>x</i>
1	readln (a, b, c, d);	1	2	3	4	
2	x:=d;	1	2	3	4	4
3	d:=c;	1	2	3	3	4
4	c:=b;	1	2	2	3	4
5	b:=a;	1	1	2	3	4
6	a:=x;	4	1	2	3	4
7	writeln (a, b, c, d);	4	1	2	3	

14. Решение задач на Паскале. Самостоятельно составьте и выполните программы решения следующих задач.

Задача 1. Среднее арифметическое. Вводятся три целых числа. Найдите среднее арифметическое этих чисел.

- 1) Дан набор команд в произвольном порядке (других команд в программе быть не должно). Соберите программу путём выбора правильных команд.

```

program sredarifm
d:=a+b+c;
writeln('sa = ', d);
d: real;
begin
var a, b, c: integer;
d:=d/3;
readln (a, b, c, d);
writeln('sa = ', sa:5:2);
var d: real;
end.
var a, b, c, d: integer;
readln (a, b, c);
var a, b, c, d: real;

```

Текст программы:



Всего команд в программе: -----



- 2) Составьте тесты для проверки правильности решения задачи.

№ теста	Входные данные			Результат
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
1				
2				
3				
4				
5				



- 3) Наберите и выполните (протестируйте) программу для предложенных тестовых данных. Если обнаружите ошибки, исправьте их.



- 4) Сделайте и сохраните скриншот решения задачи.

Задача 2. Перестановки. Вводятся две целочисленные переменные. Поменяйте местами значения переменных в ячейках.



- 1) Дан набор команд в произвольном порядке (других команд в программе быть не должно). Соберите программу путём выбора правильных команд.

```
c:=a;
program obmen
end.
writeln(a, ' ', b, ' ', c);
```

```

b:=a;
begin
var a, b: integer;
a:=c;
readln (a, b);
c:=b;
writeln(a, ' ', b);
var a, b, c: integer;
b:=c;
readln (a, b, c);
c:=a;

```

Текст программы:

```

-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----

```

Всего команд в программе: -----

- 2) Составьте тесты для проверки правильности решения задачи.



№ теста	Входные данные		Промежуточные данные	Результаты
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
1				
2				
3				
4				
5				



3) Наберите и выполните (протестируйте) программу для предложенных тестовых данных. Если обнаружите ошибки, исправьте их.

4) Сделайте и сохраните скриншот решения задачи.

Задача 3. Вычисление степени. Вводится целочисленная переменная x . Найдите x^4 за два умножения.

1) Дан набор команд в произвольном порядке (других команд в программе быть не должно). Соберите программу путём выбора правильных команд.

```
program x4
writeln(x, ' ', y);
y:=x*x;
begin
var x: integer;
x:=y*y;
readln (x, y);
end.
y:=y*y;
writeln(x);
var x, y: integer;
y:=x;
readln (x);
y:=y*x;
var x, y: real;
writeln(y);
```

Текст программы:

Всего команд в программе: -----

- 2) Составьте тесты для проверки правильности решения задачи.



№ команды	Команда	Переменная	Значения		Смысл команды
			1-й тест	2-й тест	
1	readln(x)	x	2	3	Ввод переменной
2					

- 3) Наберите и выполните (протестируйте) программу для предложенных тестовых данных. Если обнаружите ошибки, исправьте их.
- 4) Сделайте и сохраните скриншот решения задачи.



Задача 4. Вычисление степени за минимальное количество умножений. Вводится целочисленная переменная x . Найдите x^{14} за минимальное количество умножений.

- 1) Дан набор команд в произвольном порядке (других команд в программе быть не должно). Соберите программу путём выбора правильных команд.



```

program x14
y:=x*x;
writeln(x);
x:=x*y;
var x: integer;
x:=x*x;
readln(x, y);
end.
y:=y*y;
writeln(y);
y:=y*x;
```

Текст программы:

Blank handwriting practice paper with horizontal lines.

2) Составьте тесты для проверки правильности решения задачи:



№ команды	Команда	Переменная	Значения		Смысл команды
			1-й тест	2-й тест	
1	readln(x)	x	2	3	Ввод переменной
2					

№ команды	Команда	Переменная	Значения		Смысл команды
			1-й тест	2-й тест	

3) Наберите и выполните (протестируйте) программу для предложенных тестовых данных. Если обнаружите ошибки, исправьте их.

4) Сделайте и сохраните скриншот решения задачи.

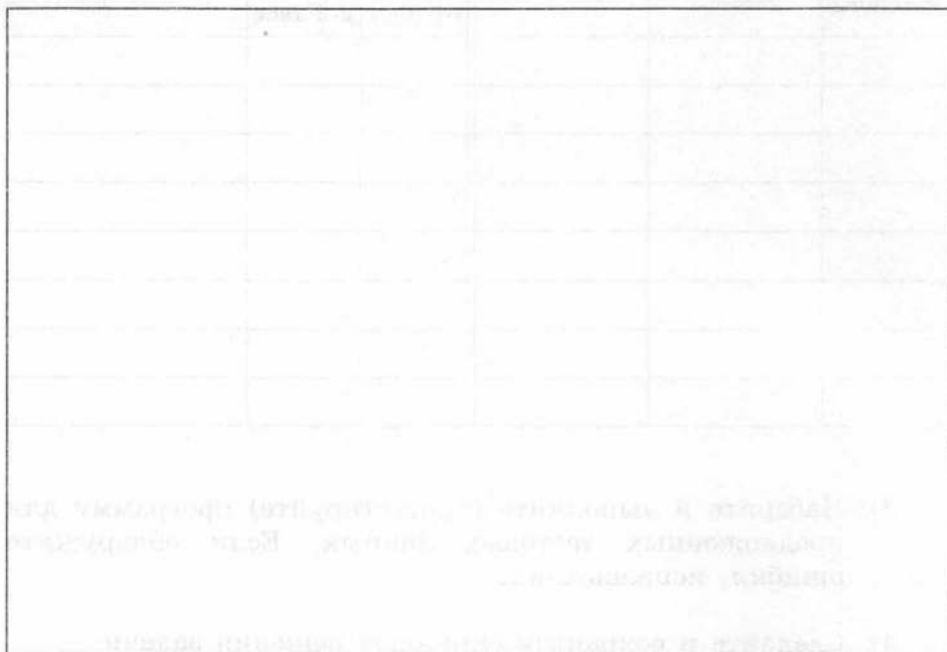
Отчёт по лабораторной работе №2

1. Распечатайте и вклейте скриншот решения задачи 1:

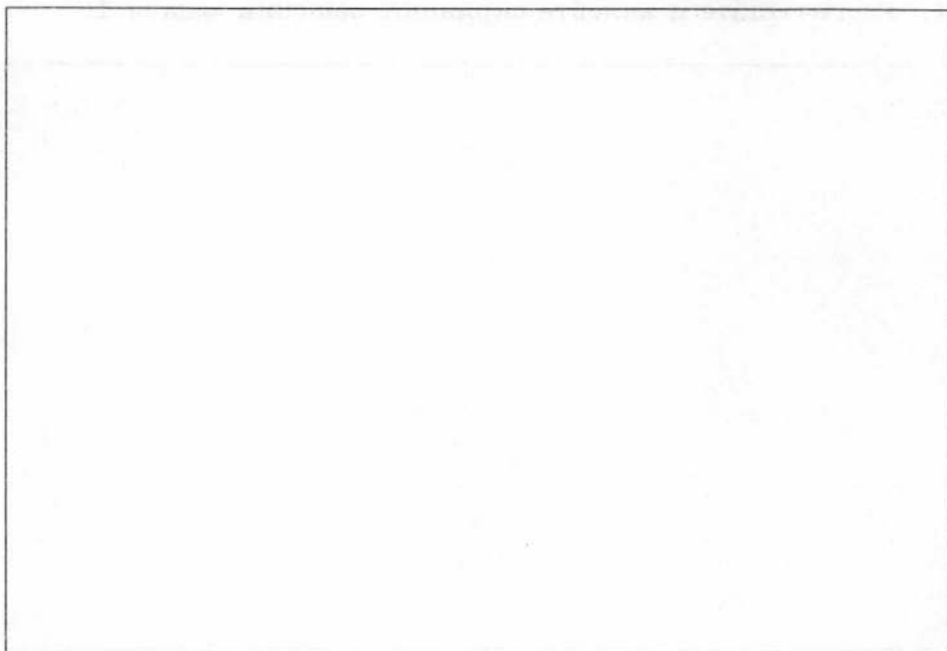




2. Распечатайте и вклейте скриншот решения задачи 2:



3. Распечатайте и вклейте скриншот решения задачи 3:



Лабораторная работа № 3

УЧИМСЯ ПРОГРАММИРОВАТЬ ЗАДАЧИ С УСЛОВИЯМИ

(К параграфу учебника: 1.2. Кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования)

Тема: изучение программирования алгоритмов с ветвлением.

Цель работы: научиться программировать задачи с условиями.

Программное обеспечение: компьютер с установленной операционной системой Windows или Linux; текстовый процессор Microsoft Word; среда программирования на языке Паскаль.



1. Какое значение будет иметь переменная z после выполнения следующего оператора?

```
if x>0 then if y>0 then z:=1 else z:=2 else z:=3;
```

При $x = 1, y = 1$ $z =$

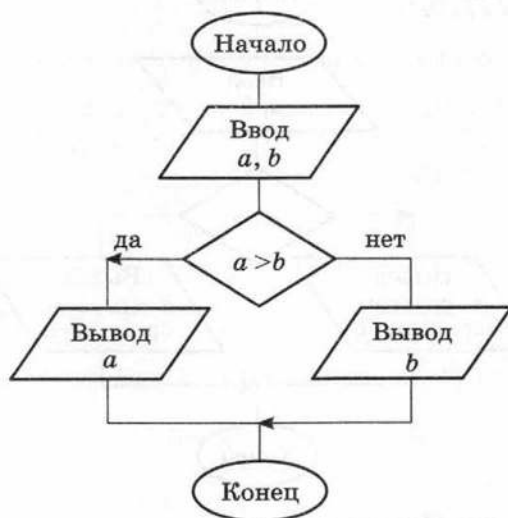
При $x = -1, y = 1$ $z =$

При $x = 1, y = -1$ $z =$



2. Примеры решения задач на Паскале. Ознакомьтесь с примерами решения задач на Паскале.

Пример 1. Вводятся два целых числа. Выведите большее из них (или их общее значение, если числа равны).

*Решение***Блок-схема:****Программа на Паскале:**

```

program max1;
var a, b: integer;
begin
  writeln ('Введите два целых числа');
  readln (a, b);
  if a>b then writeln ('max = ', a)
    else writeln ('max = ', b);
end.

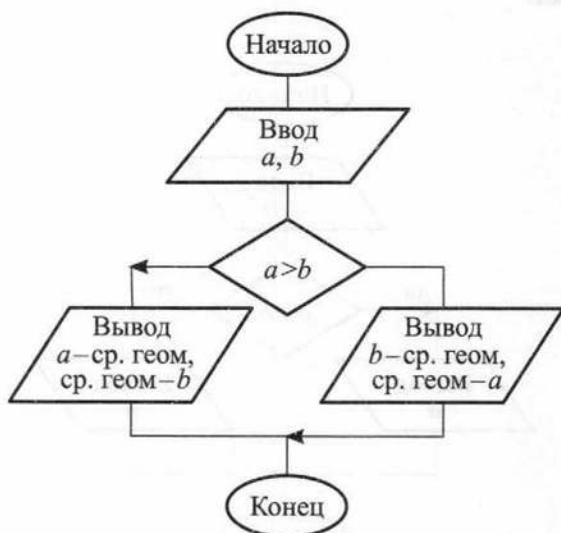
```

Пример 2. Вводятся два целых числа. Выведите: а) разность между большим числом и средним геометрическим двух чисел; б) разность между средним геометрическим двух чисел и меньшим числом.

Примечание. Среднее геометрическое двух чисел — это квадратный корень из их произведения.

Решение

Блок-схема:



Программа на Паскале:

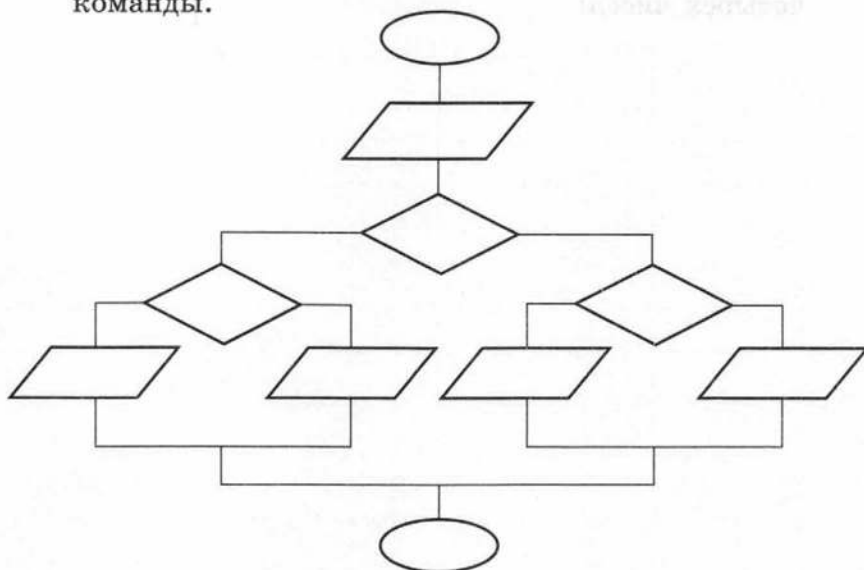
```

program raz1;
var a, b: integer;
    sg, x1, x2: real;
begin
  writeln ('Введите два целых числа');
  readln(a, b);
  sg:=sqrt(a*b);
  if a>b then begin
    x1:=a-sg;
    x2:=sg-b;
  end
  else begin
    x1:=b-sg;
    x2:=sg-a;
  end;
  writeln ('max-sg = ', x1:5:2, ' sg-min= ', x2:5:2);
end.
  
```

3. Решение задач на Паскале. Самостоятельно составьте и выполните программы решения следующих задач.

Задача 1. Поиск максимума. Вводятся три целых числа. Найдите наибольшее из этих чисел.

- 1) Вставьте в блок-схему решения задачи нужные команды.



Какое количество сравнений здесь используется?

- 2) Проведите исследование.

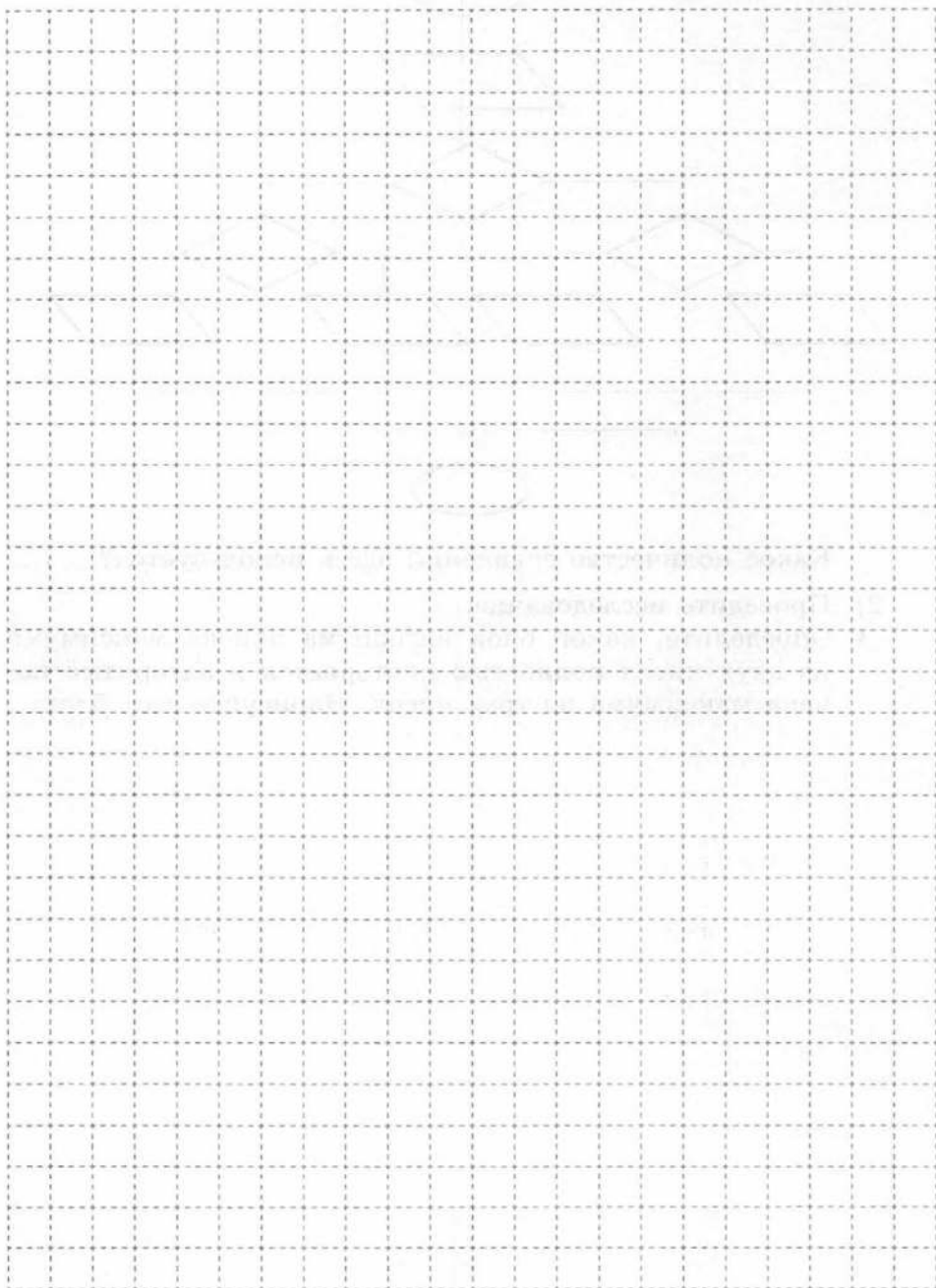
- Определите, какой блок алгоритма поиска максимума из двух чисел полностью повторяется в алгоритме поиска максимума из трёх чисел. Нарисуйте этот блок:



Сколько таких блоков в блок-схеме алгоритма поиска максимума из трёх чисел?

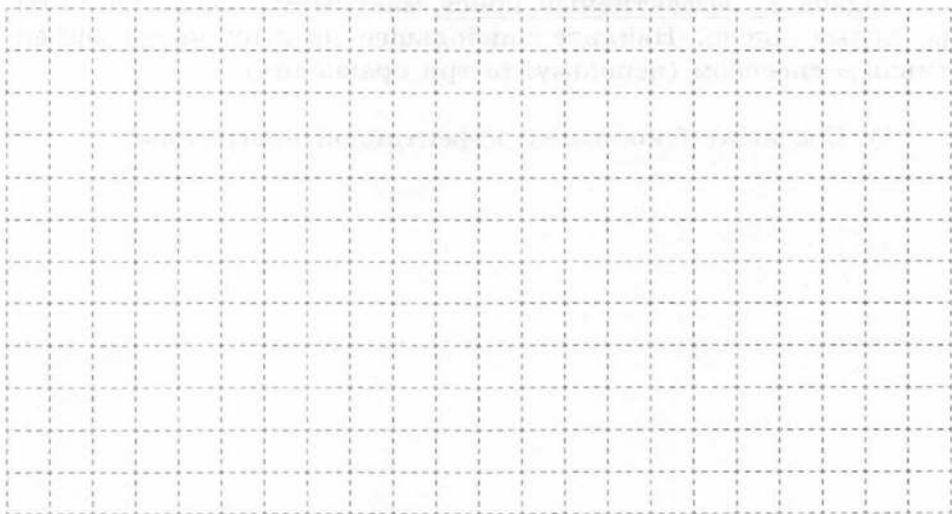


- Составьте блок-схему алгоритма поиска максимума из четырёх чисел:



Какое количество сравнений здесь используется?

- Определите, какой блок алгоритма поиска максимума из трёх чисел полностью повторяется в алгоритме поиска максимума из четырёх чисел. Нарисуйте этот блок:



Сколько таких блоков в блок-схеме алгоритма поиска максимума из четырёх чисел? -----



- Сравните блок-схемы алгоритмов поиска максимума из двух, трёх и четырёх чисел. Найдите правило (формулу) зависимости количества сравнений от количества чисел (переменных):



Заполните таблицу:

Количество чисел	Количество сравнений
2	1
3	3
4	7
5	
6	
7	
8	
9	
10	

Отметим, что этот алгоритм поиска максимума не эффективен.

Задача 2. Эффективный поиск максимума. Вводятся четыре целых числа. Найдите наибольшее из этих чисел эффективным способом (используйте три сравнения).



- 1) Составьте блок-схему эффективной программы:



- 2) Дан набор команд в произвольном порядке (других команд в программе быть не должно). Соберите программу путём выбора правильных команд.

```
max:=a;  
writeln('max = ', d);  
if c>max then  
  program max4  
  begin  
    if d<max then  
      c:=max;  
  var a, b, c, d, max: integer;  
  max:=b;  
  d:=max;
```




- 3) Составьте тесты для проверки правильности решения задачи:

№ теста	Входные переменные				Результат
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>max</i>
1					
2					
3					
4					
5					



- 4) Наберите и выполните (протестируйте) программу для предложенных тестовых данных. Если обнаружите ошибки, исправьте их.



- 5) Сделайте и сохраните скриншот решения задачи.

Задача 3. Сортировка по возрастанию. Вводятся 3 числа. Расположите их в ячейках в порядке возрастания.



- 1) Дан шаблон программы решения задачи. Дополните его — составьте программу.

```
program uporyado4it_po_vozrastaniyu;
```

```
var
```

```
  a, b, c, x: _____;
```

```
begin
```

```
  writeln('Введите 3 числа');
```

```
  readln( _____ );
```

```
  if a>b then begin
```

```
    _____;
```

```
    _____;
```

```
    _____;
```

```
  end;
```

```
if b>c then begin
```

```
-----;
```

```
-----;
```

```
-----;
```

```
end;
```

```
if ____ then begin
```

```
-----;
```

```
-----;
```

```
-----;
```

```
end;
```

```
writeln (a, ' ', b, ' ', c);
```

```
end.
```

- 2) Составьте тесты для проверки правильности решения задачи:

№ теста	Входные переменные			Результат		
	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>
1						
2						
3						
4						
5						

- 3) Наберите и выполните (протестируйте) программу для предложенных тестовых данных. Если обнаружите ошибки, исправьте их.
- 4) Сделайте и сохраните скриншот решения задачи.
- 5) Проанализируйте программу.
Почему в программе упорядочения трёх чисел нельзя обойтись двумя проверками?

```
-----
-----
-----
```



Приведите пример данных, при которых программа, имеющая только две первые проверки, даст неверный результат.

a = _____

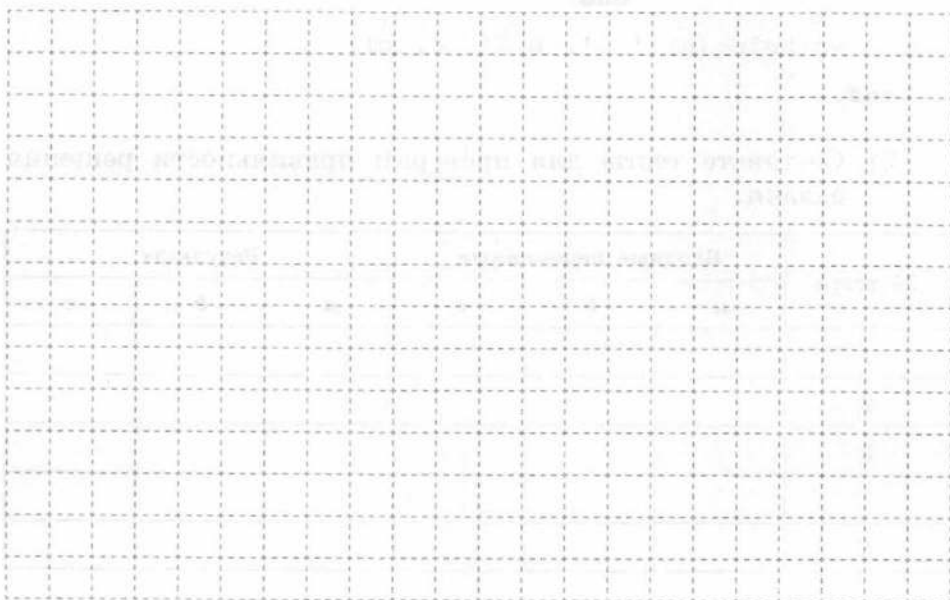
b = _____

c = _____

Для дополнительного выполнения

Задача 4. Поиск двух максимумов. Вводятся 4 числа. Выведите наибольшее и второе по величине числа.

Составьте блок-схему решения задачи:



4. Определите значение переменной c после выполнения следующего фрагмента программы (из материалов ЕГЭ).

```
a:=100;  
b:=30;  
a:=a-b*3;  
if a>b then  
    c:=a-b  
else c:=b-a;
```

Ответ: _____



5. Требовалось написать программу, которая решает уравнение $ax + b = 0$ относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно (из материалов ОГЭ):

```
program uravl
var a, b, x: real;
begin
  readln(a, b, x);
  if b=0 then
    write('x = 0')
  else
    if a=0 then
      write('Нет решений')
    else
      write('x =', -b/a);
end.
```

Последовательно выполните три задания:

- 1) Приведите пример таких чисел a , b , x , при которых программа неверно решает поставленную задачу.

- 2) Укажите, какая часть программы является лишней.

- 3) Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)



6. Требовалось написать программу, которая решает уравнение $a|x| = b$ относительно x для любых чисел a и b , введённых с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно (из материалов ГИА):

```

program urav2
var a, b, x: real;
begin
  readln(a, b, x);
  if a=0 then
    if b=0 then
      write ('любое число')
    else
      write ('нет решений')
  else
    if b=0 then
      write('x=0')
    else
      write('x =', b/a, ' или x =', -b/a);
end.

```

Последовательно выполните три задания:

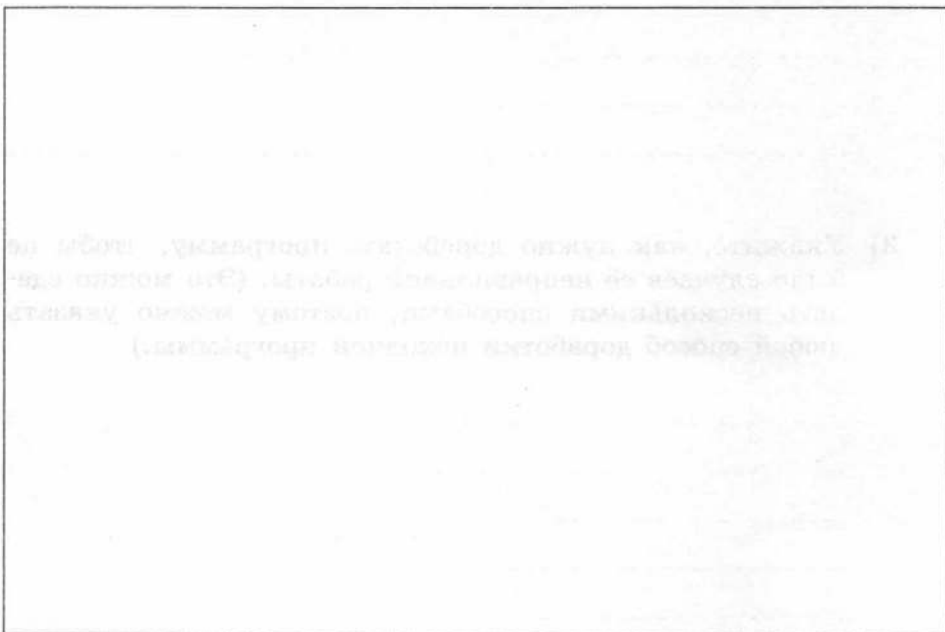
- 1) Приведите пример таких чисел a , b , x , при которых программа неверно решает поставленную задачу.

- _____
- _____
- _____
- _____
- _____

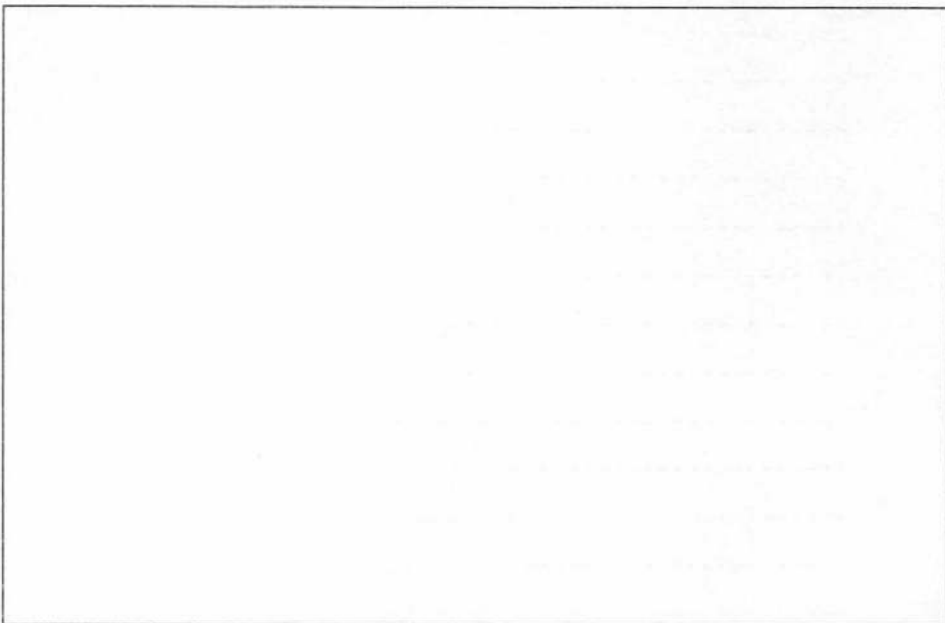
- [illegible]

Отчёт по лабораторной работе № 3

1. Распечатайте и вклейте скриншот решения задачи 2:



2. Распечатайте и вклейте скриншот решения задачи 3:



**Выводы**

Структура ветвления — базовая конструкция программирования, с её помощью можно:

От эффективности найденного алгоритма зависит:

Оценка учителя:



Лабораторная работа № 4

УЧИМСЯ ПРОГРАММИРОВАТЬ ЗАДАЧИ С ЦИКЛАМИ: ЦИКЛЫ С ПРЕДУСЛОВИЕМ И ПОСТУСЛОВИЕМ

(К параграфу учебника:

1.2. Кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования)

Тема: изучение программирования алгоритмов с циклами.

Цель работы: научиться составлять программы с циклами с предусловием (цикл «пока») и постусловием (цикл «до»).

Программное обеспечение: компьютер с установленной операционной системой Windows или Linux; текстовый процессор Microsoft Word; среда программирования на языке Паскаль.

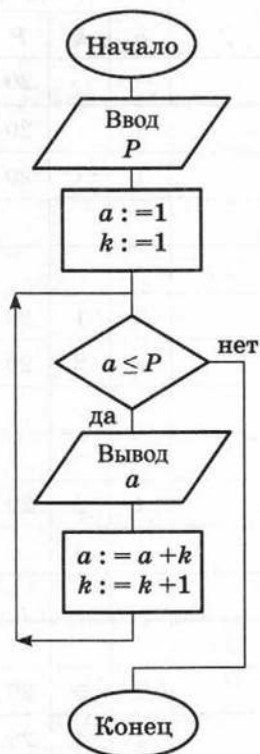


1. **Примеры решения задач на Паскале.** Ознакомьтесь с примерами решения задач на Паскале.

Пример 1. Требуется выводить числа ряда 1, 2, 4, 7, 11, 16, 22, 29 ... до тех пор, пока они не превысят заданное число P .

Решение

Блок-схема:



Программа на Паскале:

```
program ryad1;  
var a, k, P: integer;  
begin  
  readln(P);  
  a:=1;  
  k:=1;  
  while a<=P do  
    begin  
      write(a, ' ');  
      a:=a+k;  
      k:=k+1;  
    end;  
  writeln;  
end.
```

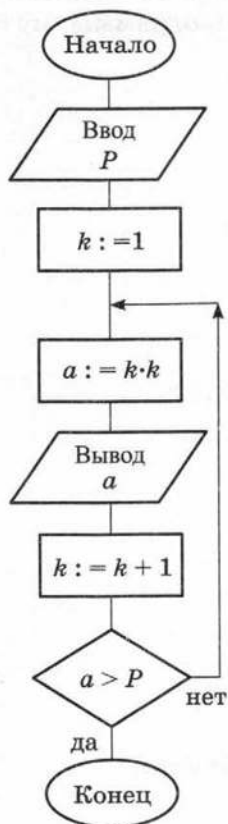
Трассировка программы:

№ команды	Команда	Значения переменных				
		<i>a</i>	<i>k</i>	<i>P</i>	Условие	Вывод
1	readln (P);			20		
2	a:=1;	1		20		
3	k:=1;	1	1	20		
4	a<=P				1 ≤ 20 (да)	
5	write (a, ' ');					1
6	a:=a+k;	2	1	20		
7	k:=k+1;	2	2	20		
8	a<=P				2 ≤ 20 (да)	
9	write (a, ' ');					2
10	a:=a+k;	4	2	20		
11	k:=k+1;	4	3			
12	a<=P				4 ≤ 20 (да)	
13	write (a, ' ');					4
14	a:=a+k;	7	3	20		
15	k:=k+1;	7	4	20		
16	a<=P				7 ≤ 20 (да)	
17	write (a, ' ');					7
18	a:=a+k;	11	4	20		
19	k:=k+1;	11	5	20		
20	a<=P				11 ≤ 20 (да)	
21	write (a, ' ');					11
22	a:=a+k;	16	5	20		
23	k:=k+1;	16	6	20		
24	a<=P				16 ≤ 20 (да)	
25	write (a, ' ');					16
26	a:=a+k;	22	6	20		
27	k:=k+1;	22	7	20		
28	a<=P				22 ≤ 20 (нет)	

Пример 2. Требуется выводить числа ряда 1, 4, 9, 16, 25, 36, 49 ... до тех пор, пока они не превысят заданное число P .

Решение

Блок-схема:



Программа на Паскале:

```
program ryad2;
var a, k: integer;
begin
  readln (P);
  k:=1;
  repeat
    a:=k*k;
    write (a, ' ');
    k:=k+1;
  until a>p;
  writeln;
end.
```

2. Решение задач на Паскале. Самостоятельно составьте и выполните программы решения следующих задач.

Для двух следующих задач дан набор команд в произвольном порядке (других команд в программе быть не должно). Соберите программы путём выбора правильных команд.

```
program chislo
b:=a div 10;
begin
writeln('k = ', k);
while a>1 do
begin
b:=b mod 10;
readln (a, b, k);
writeln('s = ', s:5:2);
s:=0;
end;
var a, k: integer;
k:=0;
a:=b mod 10;
readln (a);
writeln('s = ', s);
a:=a div 10;
var a: real;
end.
var a, b, k, s: integer;
s:=s+b;
readln (a, b);
var a, b, s: integer;
b:=a mod 10;
k:=k+1;
a:=a mod 10;
while a>0 do
```

Задача 1. Количество разрядов. Вводится целое положительное число $a \leq 109$. Определите количество разрядов в этом числе.

- 1) Соберите программу путём выбора правильных команд из представленного выше набора.



Blank handwriting practice paper with dashed lines.

2) Составьте тесты для проверки правильности решения задачи.



№ теста	Входная переменная	Результат
	а	к
1		
2		
3		
4		
5		
6		

- 3) Наберите и выполните (протестируйте) программу для предложенных тестовых данных. Если обнаружите ошибки, исправьте их.
- 4) Сделайте и сохраните скриншот решения задачи.



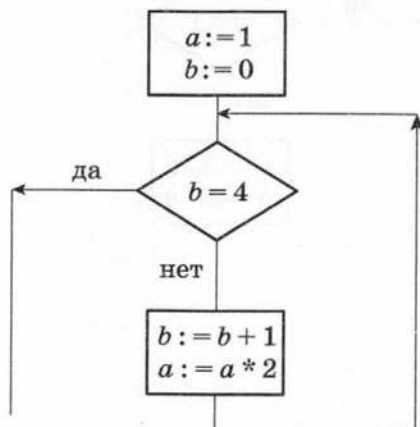


- Текст программы:

[illegible]

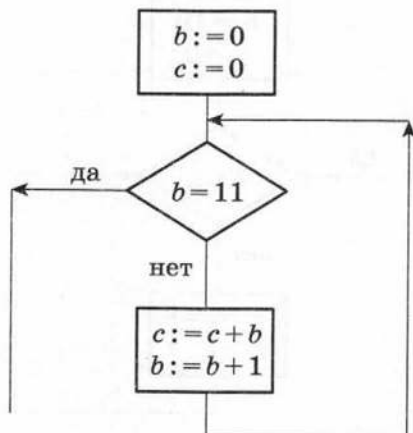
- | № теста | Входная переменная | Результат |
|---------|--------------------|-----------|
| | а | к |
| 1 | | |
| 2 | | |
| 3 | | |
| 4 | | |
| 5 | | |

- 3) Наберите и выполните (протестируйте) программу для предложенных тестовых данных. Если обнаружите ошибки, исправьте их.
 - 4) Сделайте и сохраните скриншот решения задачи.
3. Определите значение переменной a после выполнения фрагмента алгоритма, записанного в виде блок-схемы (из материалов ЕГЭ).



Ответ:

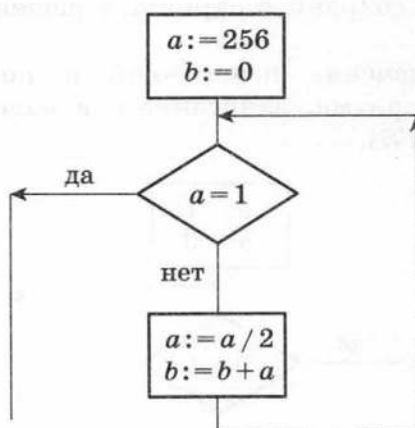
4. Определите значение переменной c после выполнения фрагмента алгоритма, записанного в виде блок-схемы (из материалов ЕГЭ).



Ответ:



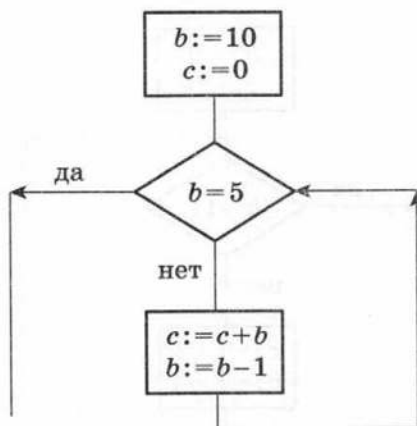
5. Запишите значение переменной b после выполнения фрагмента алгоритма, записанного в виде блок-схемы (из материалов ЕГЭ).



Ответ:



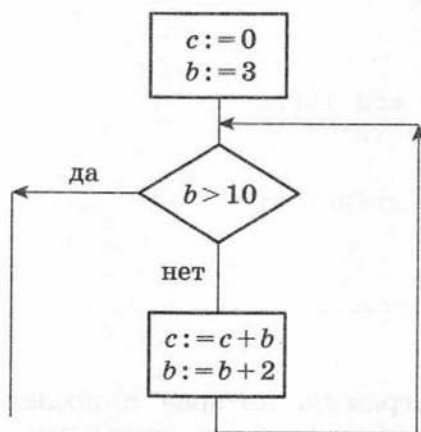
6. Определите значение переменной c после выполнения фрагмента алгоритма, записанного в виде блок-схемы (из материалов ОГЭ).



Ответ:



7. Определите значение переменной c после выполнения фрагмента алгоритма, записанного в виде блок-схемы (из материалов ОГЭ).



Ответ:

8. Определите, что будет выведено в результате выполнения программы (из материалов ЕГЭ).



```

var n, s: integer;
begin
  n:=0;
  s:=0;
  while s<=35 do
    begin
      n:=n+1;
      s:=s+4
    end;
    write(n)
  end.
  
```

Ответ:

9. Ниже записана программа. Получив на вход число x , эта программа выводит два числа: a и b . Укажите наименьшее из таких чисел x , при вводе которых программа выведет сначала 2, а потом 21 (из материалов ЕГЭ).



```
var x, a, b: integer;  
begin  
  readln(x);  
  a:=0; b:=1;  
  while x>0 do  
    begin  
      a:=a+1;  
      b:=b*(x mod 10);  
      x:=x div 10  
    end;  
  writeln(a); write(b);  
end.
```

Ответ: -----



10. Напишите программу, которая в последовательности натуральных чисел определяет сумму всех чисел, кратных 6 и оканчивающихся на 4. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 — признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000.

Введённые числа не превышают 30 000.

Программа должна вывести одно число: сумму всех чисел, кратных 6 и оканчивающихся на 4.

Пример работы программы:

Входные данные

14
24
144
22
12
0

Выходные данные

168

(Из материалов ЕГЭ.)

Дан шаблон программы решения задачи. Дополните его — составьте программу.

```
program summa_v_posledovatelnosti;  
var _____;  
begin  
    s:= _____;  
    readln ( _____ );  
    _____ a<>0 _____  
        if _____ then  
            _____;  
            _____;  
end;  
    writeln ('Кол-во чисел =' _____);  
end.
```

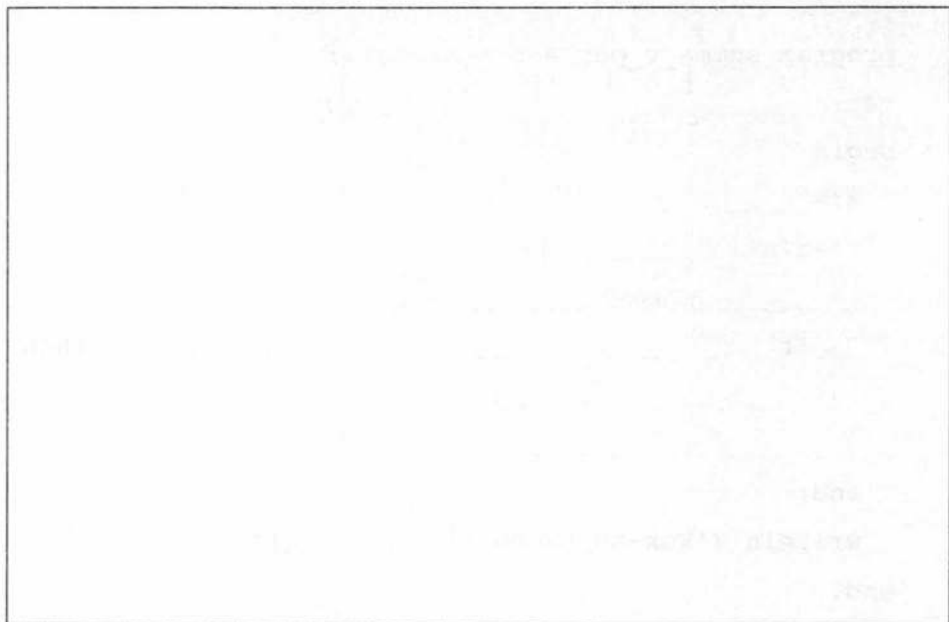
Отчёт по лабораторной работе №5

1. Распечатайте и вклейте скриншот решения задачи 1:





2. Распечатайте и вклейте скриншот решения задачи 2:



Выводы

1. Структура цикл — базовая конструкция программирования, с её помощью можно:

2. Вы изучили два вида циклов — с предусловием и с постусловием. Когда, по вашему мнению, удобнее использовать первый вид, а когда — второй?

Оценка учителя: -----

Лабораторная работа № 5

УЧИМСЯ ПРОГРАММИРОВАТЬ ЗАДАЧИ С ЦИКЛАМИ: ЦИКЛ СО СЧЁТЧИКОМ

(К параграфу учебника:

1.2. Кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования)

Тема: изучение программирования алгоритмов с циклами.

Цель работы: научиться составлять программы с циклом со счётчиком (цикл «для»).

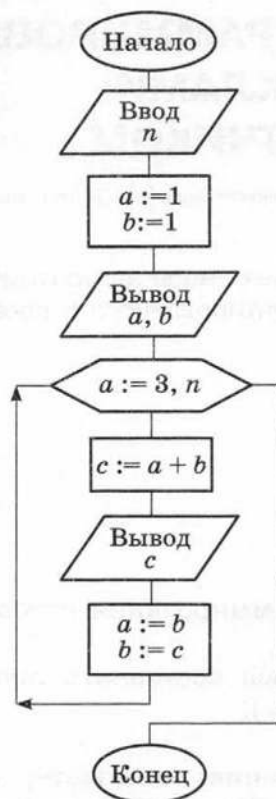
Программное обеспечение: компьютер с установленной операционной системой Windows или Linux; текстовый процессор Microsoft Word; среда программирования на языке Паскаль.

1. **Примеры решения задач на Паскале.** Ознакомьтесь с примерами решения задач на Паскале.

Пример 1. Требуется вывести n чисел ряда: 1, 1, 2, 3, 5, 8, 13, 21, 34 ... Этот ряд носит название ряда Фибоначчи и строится по следующему закону: первые два числа равны 1, каждое следующее равно сумме двух предыдущих.



Решение
Блок-схема:



Программа на Паскале:

```

program ryadfib;
var a, b, c, n, k: integer;
begin
  readln(n);
  a:=1;
  b:=1;
  write(a, ' ', b, ' ');
  for k:=3 to n do
    begin
      c:=a+b;
      write(c, ' ');
      a:=b;
      b:=c;
    end;
  writeln;
end.
  
```

Трассировка программы:

№ команды	Команда	Значения переменных						
		a	b	c	k	n	Условие	Вывод
1	readln(n);					8		
2	a:=1;	1				8		
3	b:=1;	1	1			8		
4	write (a, ' ', b, ' ');							1 1
5	for k:=3 to n do				3	8	$3 \leq 8$ (да)	
6	c:=a+b;	1	1	2	3	8		
7	write(c, ' ');							2
8	a:=b;	1	1	2	3	8		
9	b:=c;	1	2	2	3	8		
10	for k:=3 to n do				4	8	$4 \leq 8$ (да)	
11	c:=a+b;	1	2	3	4	8		
12	write(c, ' ');							3
13	a:=b;	2	2	3	4	8		
14	b:=c;	2	3	3	4	8		
15	for k:=3 to n do				5	8	$5 \leq 8$ (да)	
16	c:=a+b;	2	3	5	5	8		
17	write(c, ' ');							5
18	a:=b;	3	3	5	5	8		
19	b:=c;	3	5	5	5	8		
20	for k:=3 to n do				6	8	$6 \leq 8$ (да)	
21	c:=a+b;	3	5	8	6	8		
22	write(c, ' ');							8
23	a:=b;	5	5	8	6	8		
24	b:=c;	5	8	8	6	8		
25	for k:=3 to n do				7	8	$7 \leq 8$ (да)	
26	c:=a+b;	5	8	13	7			
27	write(c, ' ');							13

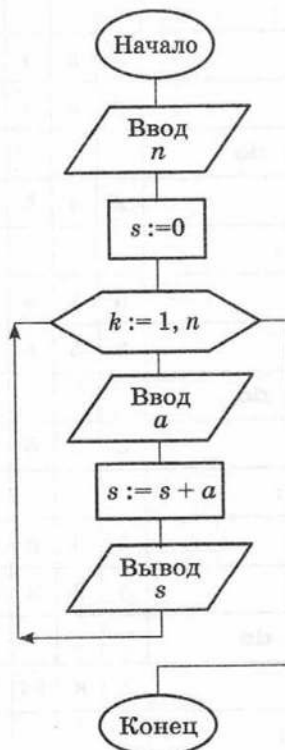
Окончание таблицы

№ команды	Команда	Значения переменных						
		a	b	c	k	n	Условие	Вывод
28	a:=b;	8	8	13	7	8		
29	b:=c;	8	13	13	7	8		
30	for k:=3 to n do				8	8	8 ≤ 8 (да)	
31	c:=a+b;	8	13	21	8	8		
32	write(c, ' ');							21
33	a:=b;	13	13	21	8	8		
34	b:=c;	13	21	21	8	8		
35	for k:=3 to n do				9	8	9 ≤ 8 (нет)	

Пример 2. С клавиатуры последовательно вводятся n чисел. Требуется после каждого ввода выводить сумму введенного числа и всех предыдущих.

Решение

Блок-схема:



Программа на Паскале:

```

program kalkulyator;
var n: integer;
    a, s: real;
begin
  readln (n);
  s:=0;
  for k:=1 to n do
    begin
      readln(a);
      s:=s+a;
      writeln ('Сумма = ', s:5:2);
    end;
end.

```

- 2. Решение задач на Паскале.** Самостоятельно составьте и выполните программу решения следующей задачи.

Задача. Требуется вывести n чисел ряда: 1, 2, 3, 6, 11, 20, 37 ...

- 1) Дан шаблон программы решения задачи. Дополните его — составьте программу.



```

program ryad;
var a, b, c, d, n, k _____;
begin
  _____(n);
  _____;
  _____;
  _____;
  _____( _____ );
  k:= _____ n _____;
  _____
  d:= _____;
  _____(d, _____);
  _____;
  _____;

```


№ команды	Команда	Значения переменных							
		a	b	c	d	k	n	Условие	Вывод
22									
23									
24									
25									
26									
27									
28									
29									
30									

3) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.

4) Сделайте и сохраните скриншот решения задачи.

3. Запишите значение переменной *s*, полученное в результате работы следующей программы (из материалов ОГЭ).

```
var s, k: integer;
begin
  s:=0;
  for k:=1 to 11 do
    s:=s+12;
  write(s);
end.
```

Ответ:

4. Запишите значение переменной *s*, полученное в результате работы следующей программы (из материалов ГИА).

```
var s, k: integer;
begin
  s:=8;
  for k:=3 to 8 do
    s:=s+8;
  writeln(s);
end.
```

Ответ:



5. Напишите эффективную программу, которая по двум данным натуральным числам a и b , не превосходящим 30 000, подсчитывает количество чётных натуральных чисел на отрезке $[a, b]$ (включая концы отрезка). Программа получает на вход два натуральных числа a и b , при этом гарантируется, что $1 \leq a \leq b \leq 30\,000$. Проверять входные данные на корректность не нужно. Программа должна вывести одно число: количество чётных чисел на отрезке $[a, b]$.

Пример работы программы:

Входные данные

10
20

Выходные данные

6

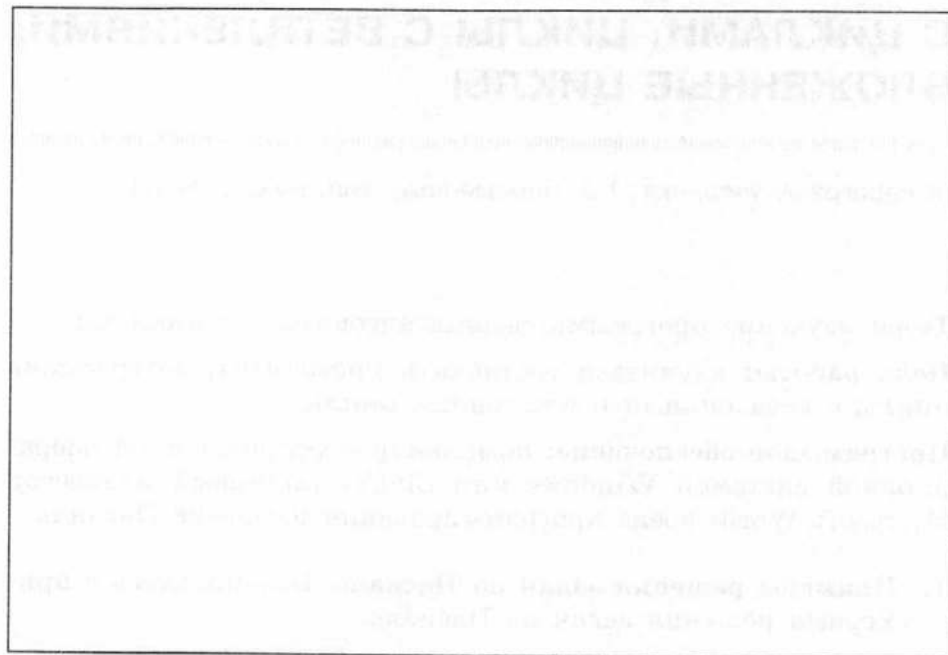
(Из материалов ГИА.)

Дан шаблон программы решения задачи. Дополните его — составьте программу.

```
program chotnie;  
var a, b, k _____;  
begin  
    _____(a, b);  
    _____;  
    if _____ then _____;  
    _____(k);  
end.
```

Отчёт по лабораторной работе № 5

1. Распечатайте и вклейте скриншот решения задачи:

**Выводы**

Когда в программировании удобно использовать цикл со счётчиком?



Оценка учителя: -----

Лабораторная работа № 6

**УЧИМСЯ ПРОГРАММИРОВАТЬ ЗАДАЧИ
С ЦИКЛАМИ: ЦИКЛЫ С ВЕТВЛЕНИЯМИ;
ВЛОЖЕННЫЕ ЦИКЛЫ**

(К параграфу учебника: 1.3. Переменные: тип, имя, значение)

Тема: изучение программирования алгоритмов с циклами.

Цель работы: научиться составлять программы, содержащие циклы с ветвлениями и вложенные циклы.

Программное обеспечение: компьютер с установленной операционной системой Windows или Linux; текстовый процессор Microsoft Word; среда программирования на языке Паскаль.

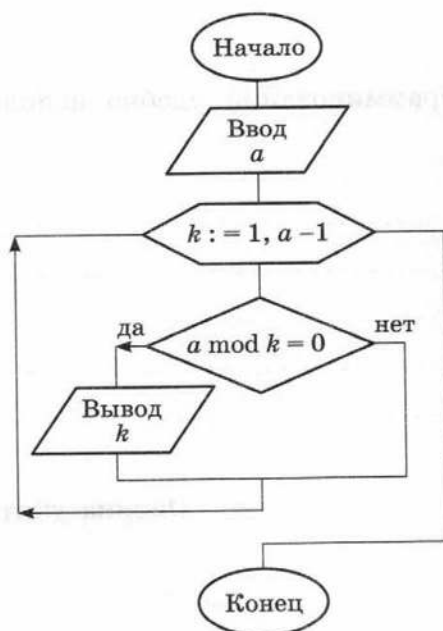


- 1. Примеры решения задач на Паскале.** Ознакомьтесь с примерами решения задач на Паскале.

Пример 1. Вводится целое число a . Требуется вывести все его делители.

Решение

Блок-схема:



Будем делить число a подряд на все числа k от 1 до $a - 1$. Если остаток равен нулю, то k — делитель числа a .

Замечание. На самом деле делить на все числа до $a - 1$ необязательно. Достаточно делить до $a/2$. Подумайте почему.

Пример 2. Требуется вывести все трёхразрядные числа-палиндромы.

Палиндромом называется такое число, слово или текст, которые одинаково читаются как слева направо, так и справа налево (примеры: 232, 5445).

Существует несколько способов решения.

Конечно, просматривать все числа от 100 до 999 и определять, является ли очередное просматриваемое число палиндромом, — крайне неоптимально.

Рассмотрим оптимальные способы решения.

Найдём закономерность в последовательности трёхразрядных чисел-палиндромов (минимальное число — 101):

101 111 121 ... 191 202 212 ... 292 ... 303 ... 999

Способ 1. Заметим, что вначале разность между очередными числами равна 10 (101, 111, ... 191) — и так девять раз, а на десятый раз разность равна 11 (191, 202). Таких групп — девять.

Программа на Паскале:

```
program palindrom-3_1_sposob;
var i, k, a: integer;
begin
  a:=101;
  for k:=1 to 9 do
    begin
      for i:=1 to 10 do
        begin
          write (a, ' ');
          a:=a+10;
        end;
        a:=a+1;
      end;
    end.
end.
```

Способ 2. Снова рассмотрим последовательность трёхразрядных чисел-палиндромов. Заметим, что если закрыть во всех числах младшую цифру (которая повторяет старшую), то оставшиеся числа дадут ряд от 10 до 99 с шагом 1. Тогда возможен следующий алгоритм.

Выполняем цикл от 10 до 99.

В теле цикла в каждом значении счётчика цикла:

выделяем старшую цифру;

дописываем её в конец числа;

выводим полученное число.

Программа на Паскале:

```
program palindrom-3_2_sposob;
var i, k, a: integer;
begin
  for i:=10 to 99 do
  begin
    k:=i div 10;
    a:=i*10+k;
    write (a, ' ');
  end;
end.
```

Способ 3. Снова закроем во всех числах младшую цифру (которая повторяет старшую). Заметим, что оставшиеся цифры меняются так: пока старшая цифра равна 1, младшая меняется от 0 до 9. После этого старшая цифра увеличивается на 1, а младшая снова меняется от 0 до 9. Так продолжается, пока старшая цифра не превосходит 9. Тогда, выделив для каждой цифры свой счётчик (цикл), мы в теле цикла будем получать нужное число и выводить его.

Программа на Паскале:

```
program palindrom-3_3_sposob;
var i, k, a: integer;
begin
  for k:=1 to 9 do
    for i:=0 to 9 do
      begin
        a:=k*100+i*10+k;
        write (a, ' ');
      end;
  end;
end.
```

Заметим, что способы 2 и 3 проще способа 1.

При увеличении числа разрядов в палиндроме способ 3 оказывается проще способа 2.

2. Решение задач на Паскале. Самостоятельно составьте и выполните программы решения следующих задач.

Задача 1. Требуется вывести все четырёхразрядные числа-палиндромы описанными в примерах способами 2 и 3.

Способ 2

- 1) Дан шаблон программы решения задачи. Дополните его — составьте программу.



```

program palindrom4_2_sposob;
var
    a, b, c, k _____;
begin
    _____ k:= _____
    _____
    b:= _____;
    c:= _____;
    a= k * _____ + b * _____ + c ;
    _____ (a, _____ );
    _____;
end.
```

- 2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.
- 3) Сделайте и сохраните скриншот решения задачи.



Способ 3

- 1) Дан шаблон программы решения задачи. Дополните его — составьте программу.



```

program palindrom4_3_sposob;
var
    a, k, m _____;
begin
    _____ k:= _____
```

```

----- m:= -----
-----
----- a:= -----;
----- (a, -----);
-----;
end.

```

- 2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.
- 3) Сделайте и сохраните скриншот решения задачи.

Используя представленный набор команд, составьте и протестируйте программу решения одной из следующих задач — либо задачи 2, либо задачи 3, по согласованию с учителем.

Задача 2. Требуется вывести все числа a из интервала от 10 до 99, квадрат которых оканчивается этим же числом.

Задача 3. Требуется вывести все совершенные числа a из интервала от 1 до N .

Совершенными называются такие числа, сумма делителей которых (исключая само число) равна самому числу.

Примеры: $4 \neq 1 + 2$ — не совершенное число;
 $6 = 1 + 2 + 3$ — совершенное число;
 $8 \neq 1 + 2 + 4$ — не совершенное число.

- 1) Дан набор команд в произвольном порядке (других команд в программе быть не должно). Соберите программу путём выбора правильных команд.

```

program 2-3
for a:=1 to n do
end;
begin
writeln(a);
b:=b+c;
begin
if a mod b = 0 then;
if b=c then
c:=b mod 100;
var a, b, c: integer;

```

[illegible]



2) Наберите и выполните программу (для задачи 3 — для небольшого значения N). Если обнаружите ошибки, исправьте их.

3) Сделайте и сохраните скриншот решения задачи.

Для задачи 3:

Найденные совершенные числа:

Подумайте, что нужно изменить в программе для поиска совершенных чисел в большом интервале (например, при $N = 100\,000$ или больше).

Для дополнительного выполнения

Задача 4

а) Требуется вывести все пятиразрядные числа-палиндромы способом 3.

1) Дан шаблон программы решения задачи. Дополните его — составьте программу.

```
program palindrom5_3;
```

```
var
```

```
  a, i, j, k -----;
```

```
begin
```

```
  ----- i:= -----
```

```
  ----- j:= -----
```

```
  ----- k:= -----
```

```
  -----  
  a:= -----;
```

```

        _____(a, _____);
        _____;

end.

```

2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.



3) Сделайте и сохраните скриншот решения задачи.



б) Требуется вывести все шестизначные числа-палиндромы способом 3.

1) Дан шаблон программы решения задачи. Дополните его — составьте программу.



```

program palindrom6_3;
var
    a, i, j, k_____;
begin
    _____ i:= _____;
    _____ j:= _____;
    _____ k:= _____;
    _____
    a:= _____;
    _____ (a, _____);
    _____;
end.

```

2) Наберите и выполните программу. Если обнаружите ошибки, исправьте их.

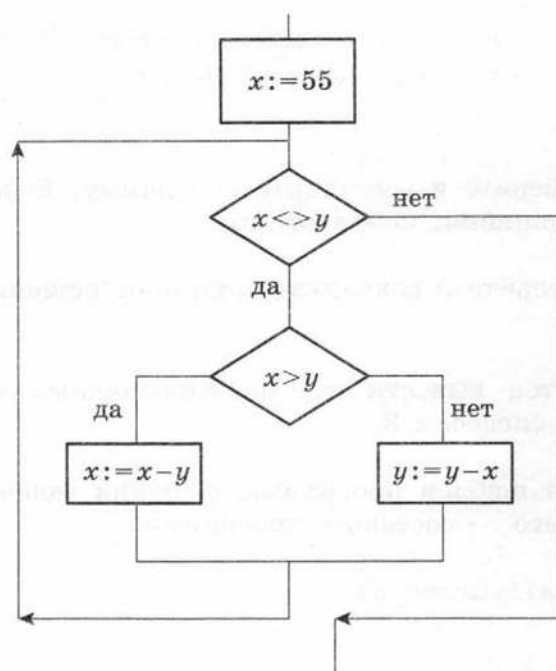


3) Сделайте и сохраните скриншот решения задачи.



3. Определите значение целочисленной переменной x после выполнения фрагмента алгоритма (из материалов ОГЭ).

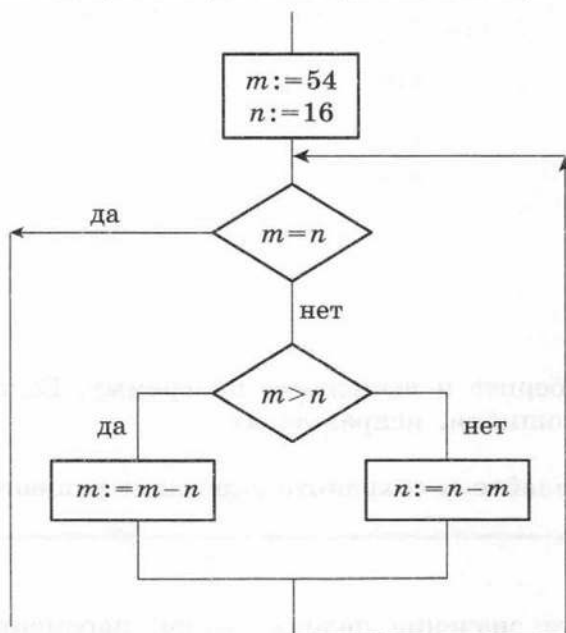




Ответ:

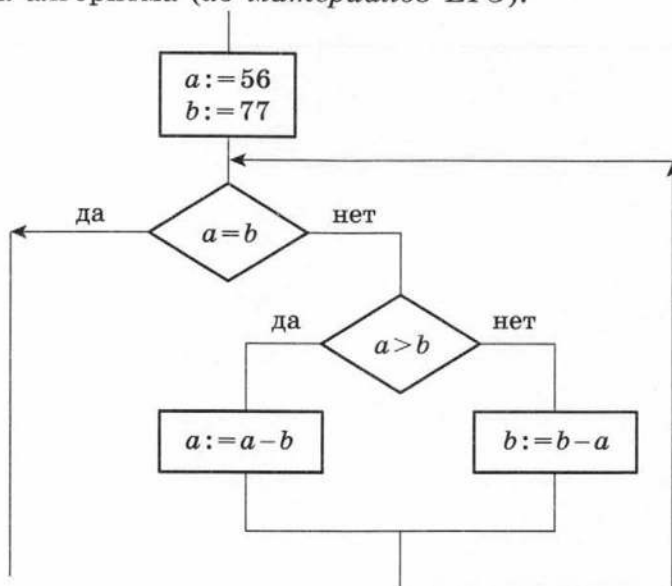


4. Определите значение переменной m после выполнения фрагмента алгоритма (из материалов ЕГЭ).



Ответ:

5. Запишите значение переменной a после выполнения фрагмента алгоритма (из материалов ЕГЭ).



Ответ:

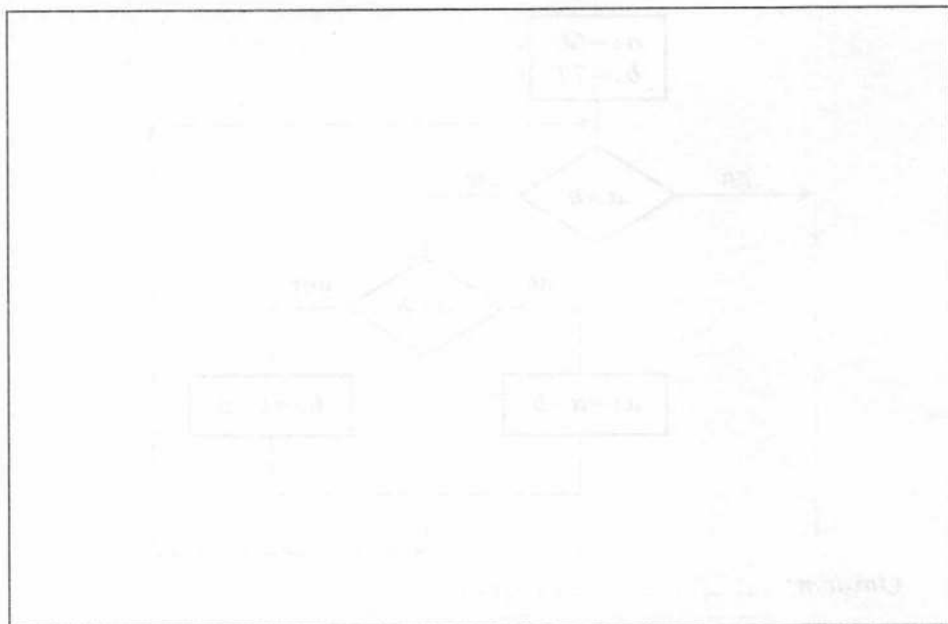
Отчёт по лабораторной работе № 6

1. Распечатайте и вклейте скриншот решения задачи 1 (способ 2):

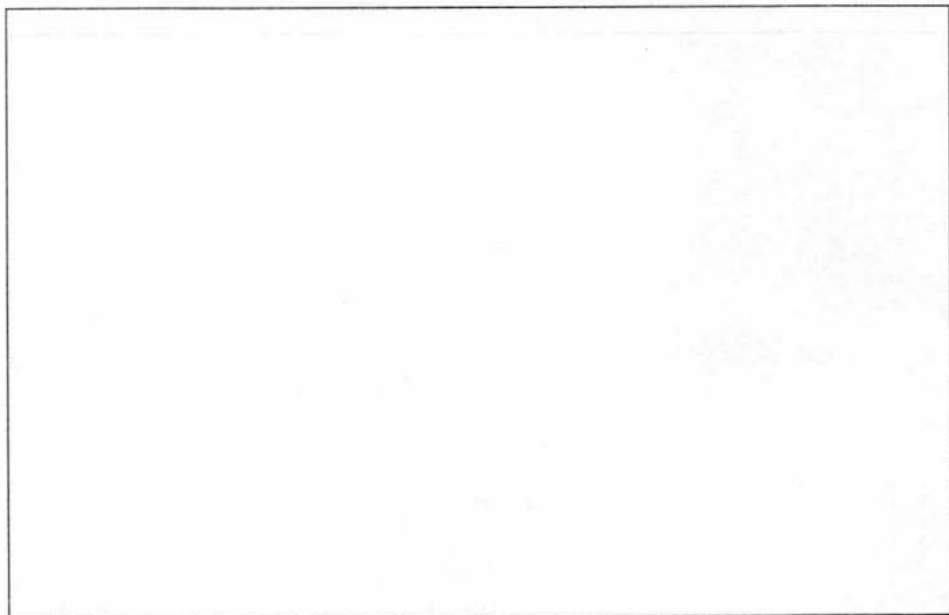




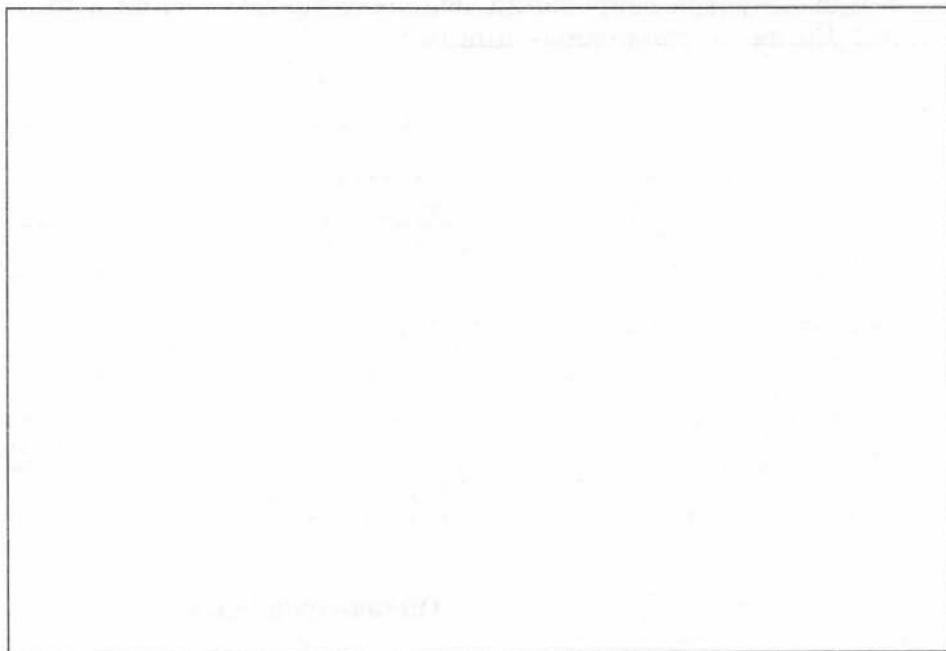
2. Распечатайте и вклейте скриншот решения задачи 1 (способ 3):



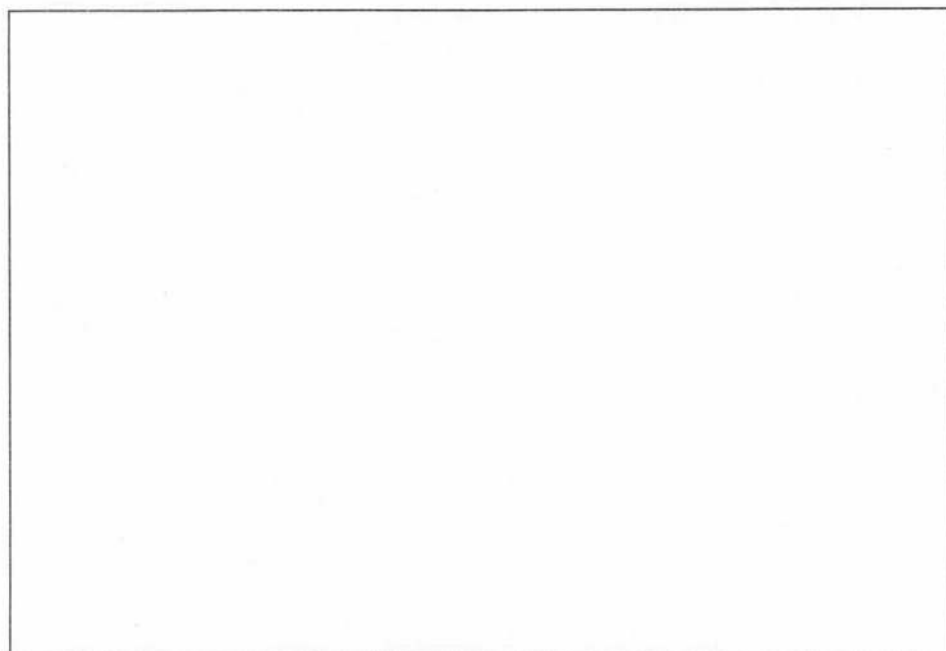
3. Распечатайте и вклейте скриншот решения задачи 2 или 3:



4. Распечатайте и вклейте скриншот решения задачи 4, а:



5. Распечатайте и вклейте скриншот решения задачи 4, б:



**Выводы**

Когда в программировании используется ветвление в цикле? Когда — вложенные циклы?

Оценка учителя: -----

Лабораторная работа № 7

УЧИМСЯ ПРОГРАММИРОВАТЬ ЗАДАЧИ СО СЛОЖНЫМИ УСЛОВИЯМИ

(К параграфу учебника:

1.4. Арифметические, строковые и логические выражения)

Тема: изучение программирования алгоритмов со сложными условиями.

Цель работы: научиться программировать задачи со сложными условиями.

Программное обеспечение: компьютер с установленной операционной системой Windows или Linux; текстовый процессор Microsoft Word: среда программирования на языке Паскаль.

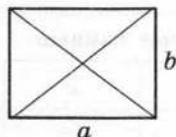
1. Примеры решения задачи на Паскале. Ознакомьтесь с примерами решения задач на Паскале.



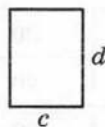
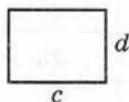
Пример 1. Вводятся размеры конверта — $a \times b$ и открытки — $c \times d$. Определите, поместится ли открытка в конверт.

Решение

Вспомним, что конверты бывают такой формы:

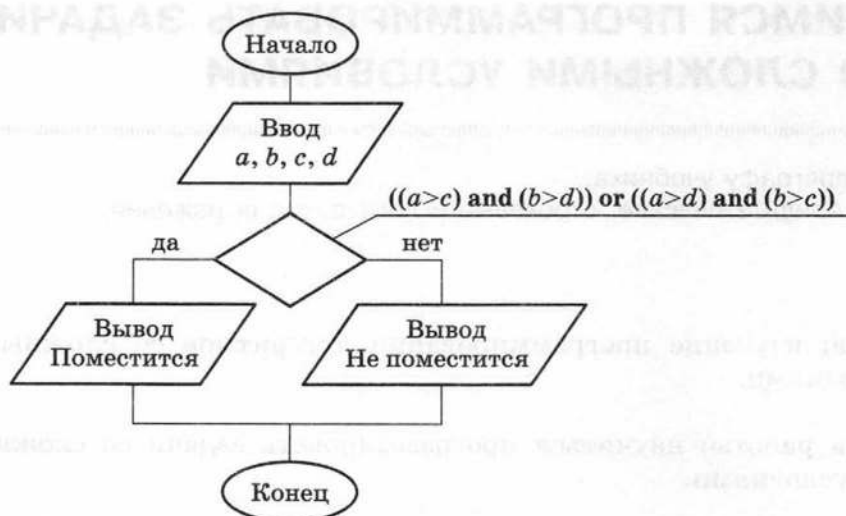


открытки бывают двух видов:



Поэтому надо проверить оба случая.

Блок-схема:



Программа на Паскале:

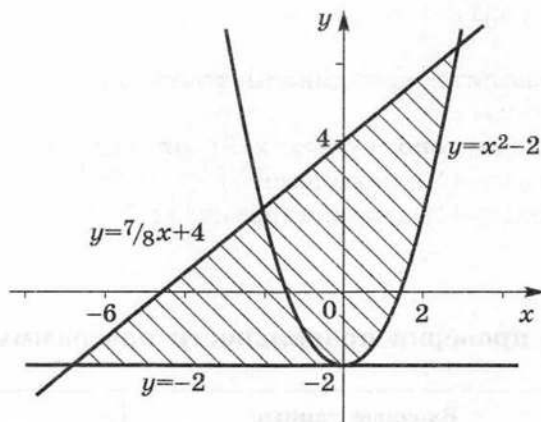
```

program konvert;
var a, b, c, d: integer;
begin
  writeln ('Введите размеры конверта и открытки');
  readln (a, b, c, d);
  if ((a>c) and (b>d)) or ((a>d) and (b>c)) then
    writeln ('Поместится')
  else writeln ('Не поместится');
end.
  
```

Тесты для проверки правильности программы:

№ теста	Входные данные				Результат
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	
1	20	20	15	15	Поместится
2	30	20	25	15	Поместится
3	30	20	15	25	Поместится
4	30	20	25	25	Не поместится
5	30	20	0	0	Поместится
6	30	20	30	19	Не поместится

Пример 2. Определите, принадлежит ли точка A с координатами (x_a, y_a) заштрихованной области.



Решение

Чтобы определить, принадлежит ли точка заштрихованной области, разделим область на две части:

- 1) внутри параболы $y = x^2 - 2$, но ниже наклонной прямой $y = \frac{7}{8}x + 4$;
- 2) внутри треугольника, ограниченного осью координат OY , прямой $y = -2$, параллельной оси OX , и наклонной прямой $y = \frac{7}{8}x + 4$.

Точка принадлежит заштрихованной области, если она находится хотя бы в одной из описанных частей.

Первая часть описывается условием:

$$(y_a > x^2 - 2) \text{ and } (y_a < \frac{7}{8}x_a + 4).$$

Вторая часть описывается условием:

$$(y_a > -2) \text{ and } (y_a < \frac{7}{8}x_a + 4) \text{ and } (x_a < 0).$$

Условие принадлежности точки всей области:

$$((y_a > x^2 - 2) \text{ and } (y_a < \frac{7}{8}x_a + 4)) \text{ or } ((y_a > -2) \text{ and } (y_a < \frac{7}{8}x_a + 4) \text{ and } (x_a < 0)).$$

Можно преобразовать выражение:

$$(y_a < \frac{7}{8}x_a + 4) \text{ and } ((y_a > x^2 - 2) \text{ or } ((y_a > -2) \text{ and } (x_a < 0))).$$

Программа на Паскале:

```

program prinadl_tochki;
var  xa, ya: real;
begin
  writeln ('Введите координаты точки');
  readln (xa, ya);
  if (ya<7/8*xa+4) and ((ya>x*x-2) or ((ya>-2) and (xa<0)))
    then writeln ('Принадлежит')
    else writeln ('Не принадлежит');
end.

```

Тесты для проверки правильности программы:

№ теста	Входные данные		Результат
	xa	ya	
1	-5	4	Не принадлежит
2	1	7	Не принадлежит
3	-1	0	Принадлежит
4	1	2	Принадлежит
5	-1	-3	Не принадлежит
6	1	-1	Не принадлежит

2. Решение задач на Паскале. Самостоятельно составьте и выполните программы решения следующих задач.

Задача 1. Кирпич. Вводятся размеры кирпича $a \times b \times c$ и отверстия $x \times y$. Установите, пройдёт ли кирпич в отверстие. (Неизвестно, какая из величин x и y больше и как распределяются значения величин a , b и c .)

- 1) Опишите на русском языке оптимальный алгоритм решения задачи крупнёнными блоками:





- 2) Дан шаблон программы решения задачи. Дополните его — составьте программу.

```

program kirpitch;
var
  a, b, c, x, y, p: _____;
begin
  writeln ('Введите размеры отверстия:');
  readln ( _____ );
  writeln ('Введите размеры кирпича:');
  readln ( _____ );
  {Расположим длины сторон отверстия в порядке убывания}
  if x<y then _____
    _____;
    _____;
    _____;
    _____;
  {Расположим длины сторон кирпича в порядке убывания}
  if a<b then begin
    _____;
    _____;
    _____;
  end;
  if b<c then begin
    _____;
    _____;
    _____;
  end;
  if _____ then begin
    _____;
    _____;
    _____;
  end;
end;

```

{Теперь достаточно, чтобы две меньшие стороны кирпича были меньше сторон отверстия. Подумайте почему}

```
if ( _____ ) then writeln ('Пройдет')
    else writeln ('Не пройдет');

end.
```



- 3) Составьте тесты для проверки правильности решения задачи:

№ теста	Входные данные					Результат (пройдёт/нейдёт)
	Отверстие		Кирпич			
	<i>x</i>	<i>y</i>	<i>a</i>	<i>b</i>	<i>c</i>	
1						
2						
3						
4						
5						
6						
7						
8						

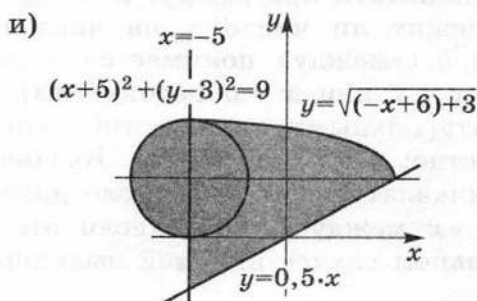
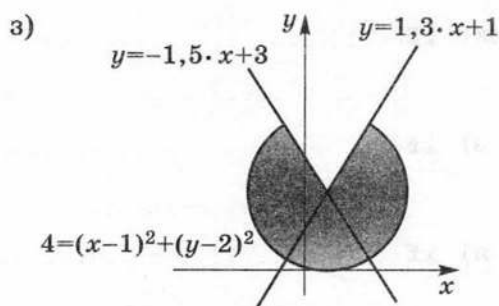
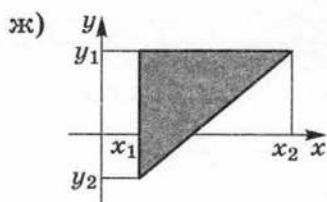
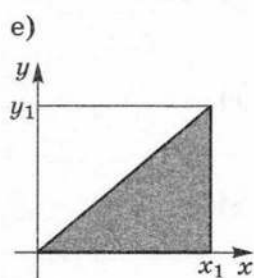
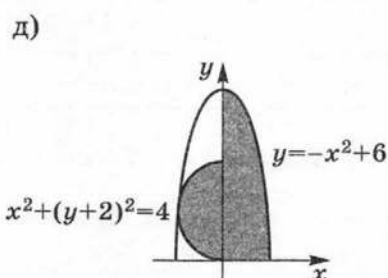
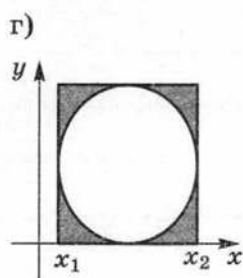
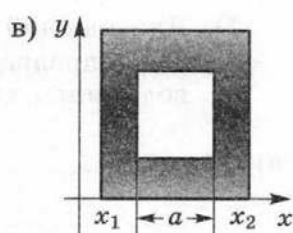
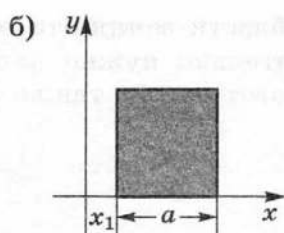
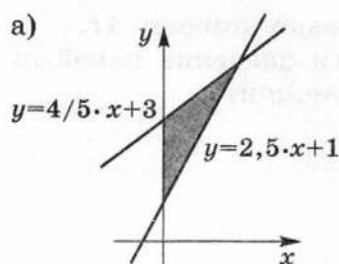


- 4) Наберите и выполните (протестируйте) программу для предложенных тестовых данных. Если обнаружите ошибки, исправьте их.



- 5) Сделайте и сохраните скриншот решения задачи.

Задача 2. Точка внутри области. Вводятся параметры геометрической фигуры (показаны на рисунках) и координаты точки $A(x_a, y_a)$. Принадлежит ли точка A закрашенной области?





- 1) Для каждой области запишите только команду **if**.
Если предварительно нужно найти значение какой-то величины, то этот расчёт также запишите.

а) **if**

.....

б) **if**

.....

в) **if**

.....

г) **if**

.....

д) **if**

.....

е) **if**

.....

ж) **if**

.....

з) **if**

.....

и) **if**

.....

3. Требовалось написать программу, в которой нужно было проверить, лежит ли число x на числовой оси между числами a и b («между» понимается в строгом смысле, т. е. случай $x = a$ или $x = b$ недопустим). Числа x , a , b являются натуральными, и известно, что a отлично от b (но неизвестно: $a > b$ или $b > a$). Входная информация вводится с клавиатуры, а на выходе должно быть сообщение вида « x между a и b » (если это действительно так), в противном случае никакой выходной информации не выдаётся.

Программист торопился и написал программу неправильно (из материалов ОГЭ):

```
var a, b, x: integer;  
    p: integer;  
begin  
    readln(a, b, x);  
    if (a>x) and (x>b) then  
        writeln ('x между a, b');  
end.
```

Последовательно выполните три задания.

- 1) Приведите пример таких чисел a , b , x , при которых программа неверно решает поставленную задачу.

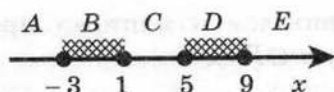


- 2) Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)



- 3) Укажите, как можно доработать программу, соблюдая дополнительное условие: доработанная программа не должна использовать логических операций **and** и **or**.

4. Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой (x — действительное число) и определяется принадлежность этой точки одному из выделенных отрезков B и D (включая границы).



Программист торопился и написал программу неправильно (из материалов ОГЭ):

```
begin
  readln(x);
  if x >= -3 then
    if x <= 9 then
      if x > 1 then
        write('Не принадлежит')
      else
        write('Принадлежит')
    end.
end.
```

Последовательно выполните два задания.



- 1) Заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D и E). Границы (точки -3, 1, 5 и 9) принадлежат заштрихованным областям (B и D соответственно).

Область	Условие 1 $x \geq -3$	Условие 2 $x \leq 9$	Условие 3 $x > 1$	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					

В столбцах условий укажите «Да», если условие выполняется; «Нет», если условие не выполняется; «—» (прочерк), если условие не будет проверяться; «не изв.», если программа ведёт себя по-разному для разных значений, принадлежащих данной области.

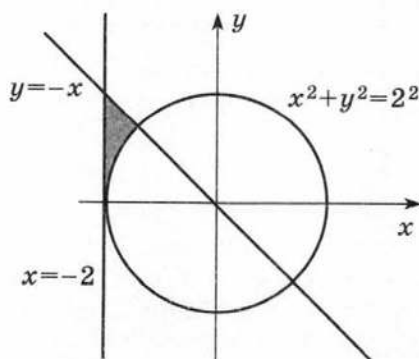
В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, введите «—» (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.».

В последнем столбце укажите «Да» или «Нет».

- 2) Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)



5. Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y — действительные числа) и определяется принадлежность этой точки заданной заштрихованной области (включая границы). Программист торопился и написал программу неправильно (из материалов ГИА).



```
var x, y: real;  
begin  
  readln(x, y);  
  if x*x+y*y>=4 then  
    if x>=-2 then  
      if y<=-x then  
        write('Принадлежит')  
      else  
        write('Не принадлежит')  
    end.  
end.
```

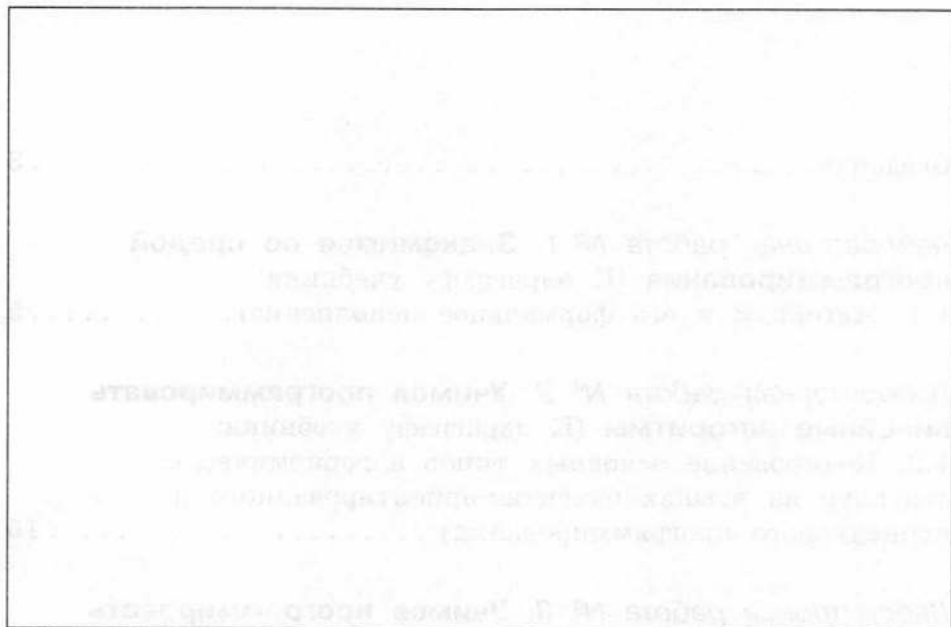
Последовательно выполните два задания.

- 1) Приведите пример таких чисел x , y , при которых программа неверно решает поставленную задачу:

- 2) Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

Отчёт по лабораторной работе № 7

1. Распечатайте и вклейте скриншот решения задачи 1:



Выводы



Логические операции в языках программирования удобно использовать для:

В каких случаях, по вашему мнению, удобнее обойтись без логических операций?

Оценка учителя: _____

ОГЛАВЛЕНИЕ

Введение	3
----------------	---

Лабораторная работа № 1. Знакомимся со средой программирования (К параграфу учебника: 1.1. Алгоритм и его формальное исполнение)	5
---	---

Лабораторная работа № 2. Учимся программировать линейные алгоритмы (К параграфу учебника: 1.2. Кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования)	15
--	----

Лабораторная работа № 3. Учимся программировать задачи с условиями (К параграфу учебника: 1.2. Кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования)	30
--	----

Лабораторная работа № 4. Учимся программировать задачи с циклами: циклы с предусловием и постусловием (К параграфу учебника: 1.2. Кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования)	46
---	----

Лабораторная работа № 5. Учимся программировать задачи с циклами: цикл со счётчиком (К параграфу учебника: 1.2. Кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования)	59
--	----

Лабораторная работа № 6. Учимся программировать задачи с циклами: циклы с ветвлениями; вложенные циклы (К параграфу учебника: 1.3. Переменные: тип, имя, значение).....	68
--	-----------

Лабораторная работа № 7. Учимся программировать задачи со сложными условиями (К параграфу учебника: 1.4. Арифметические, строковые и логические выражения).....	81
--	-----------

Учебное издание

**Угрюмович Николай Дмитриевич
Серёгин Игорь Александрович
Полежаева Ольга Александровна**

ИНФОРМАТИКА

9 класс

Рабочая тетрадь

В 2 частях

Часть 1

*Научный редактор М. С. Цветкова
Ведущий редактор О. А. Полежаева
Художник Н. А. Новак
Технический редактор Е. В. Денюкова
Корректор Е. Н. Клитина
Компьютерная верстка: Л. В. Катуркина*

Подписано в печать 27.12.16. Формат 70х100/16. Усл. печ. л. 7,8.
Тираж 3000 экз. Заказ № 39641.

ООО «БИНОМ. Лаборатория знаний»
127473, Москва, ул. Краснопролетарская, д. 16, стр. 1,
тел. (495)181-53-44, e-mail: binom@Lbz.ru
<http://www.Lbz.ru>, <http://metodist.Lbz.ru>

Отпечатано в соответствии с качеством предоставленных издательством
электронных носителей в АО «Саратовский полиграфкомбинат».
410004, г. Саратов, ул. Чернышевского, 59. www.sarpk.ru