5. Алгоритмизация и программирование

Основные понятия

Тема «Алгоритмизация и программирование» наиболее широко представлена в ЕГЭ по информатике. Для успешного выполнения заданий этой темы нужно знать и уметь использовать на практике основные алгоритмические конструкции:

- следование;
- полное и неполное ветвление;
- цикл с предусловием;
- цикл с постусловием;
- цикл с заданным числом итераций;
- вспомогательный алгоритм (подпрограмма).

Необходимо уметь составлять, анализировать и выполнять алгоритмы, используя различные формы записи:

- естественный язык;
- графический язык (блок-схемы);
- формальный язык (язык программирования).

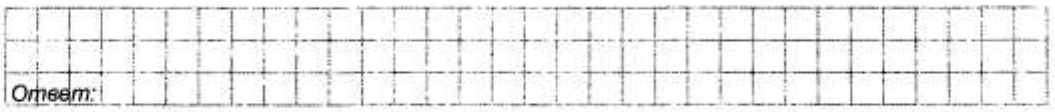
Требуется также уметь использовать простые структуры данных — одномерные и двумерные массивы.

Практическая часть

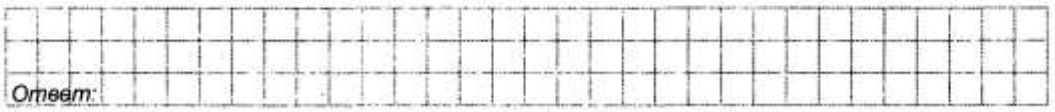
5.1. Программа для некоторого арифметического устройства представляет собой последовательность символов «+» и « · ». Символ «+» означает прибавление единицы к текущему результату вычислений, « · » — умножение на 2. Выполнение программы производится слева направо. Найдите результат работы устройства для заданного исходного числа и программы:

а) 3 +++ · · +

б) 0 · · ++ · ·



в) −1 · ++ · ·



г) −2008 ++ ·

Ответ:

5.2. Составьте для устройства, описанного в предыдущем задании, программу, длиной не более 6 символов, преобразующую заданное исходное число в заданный результат:

а) из 0 получить 17.

Ответ: _____.

б) из 3 получить 30.

Ответ: _____.

в) из −5 получить −30.

Ответ: _____.

г) из −2 получить 5.

Ответ: _____.

Рассмотрим пример решения задачи с исполнителем РОБОТ

Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо

При выполнении этой команды РОБОТ перемещается на соответствующую клетку.

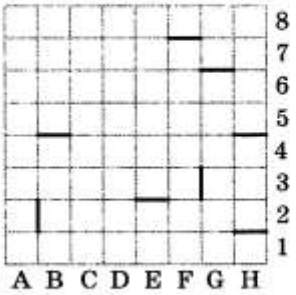
Команды проверки истинности условия на наличие стены у той клетки, где он находится:

сверху свободно	снизу свободно	слева свободно	справа свободно

Если РОБОТ начнет движение в сторону стены, то он разрушится.

а) Сколько клеток данного лабиринта соответствуют требованию, что, выполнив предложенную программу, РОБОТ остановится в той же клетке, с которой он начал движение?

НАЧАЛО
ПОКА **справа свободно**
    ДЕЛАТЬ **вправо**
ПОКА **снизу свободно**
    ДЕЛАТЬ **вниз**
ПОКА **слева свободно**
    ДЕЛАТЬ **влево**
ПОКА **сверху свободно**
    ДЕЛАТЬ **вверх**
КОНЕЦ



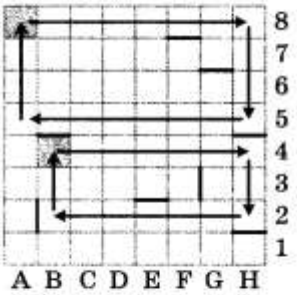
В ответе запишите число — количество таких клеток, а далее, через запятые, их адреса (сначала идет латинская буква столбца, а затем цифра строки).
Например, нижний левый угол лабиринта имеет адрес A1.

Решение

Сначала определим свойства клеток, отвечающих требованию задачи.
Поскольку последний цикл программы РОБОТА

ПОКА сверху свободно
ДЕЛАТЬ вверх

искомые клетки должны иметь границу сверху.
Таких клеток 14: вся верхняя горизонталь и клетки F7, G6 , B4, H4, E2 и H1.



Предпоследний цикл программы РОБОТА

ПОКА слева свободно
ДЕЛАТЬ влево

Это значит, что в вертикали искомой клетки ниже нее обязательно должна быть клетка с левой границей, от которой «отскочил» РОБОТ перед тем, как пойти вверх.
Из ранее найденных клеток этому условию удовлетворяют 4 — A8, B8, B4, G6
При этом B8 надо тоже отбросить, поскольку в ее вертикали на пути от клетки B2 стоит горизонтальная граница между B4 и B5.
Далее используем остальные фрагменты программы. Выполнив программу целиком для трех оставшихся клеток, получаем, что условию задачи удовлетворяют только две из них — A8 и B4
Ответ: 2, A8, B4.

5.3. Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.
Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	снизу свободно	слева свободно	справа свободно

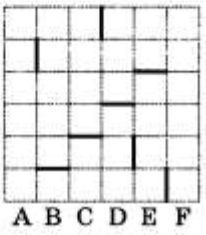
Цикл

ПОКА < *условие* > команда

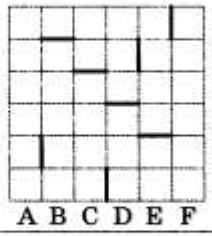
выполняется, пока условие истинно, иначе происходит переход на следующую строку. Если РОБОТ начнет движение в сторону стены, то он разрушится и программа прервется.

Нарисуйте путь робота в лабиринте, если он начал движение по программе из указанной клетки.

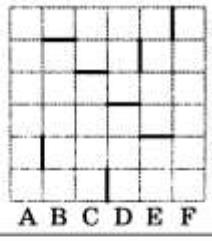
а)

Лабиринт	Программа	Начальная клетка
	НАЧАЛО ПОКА < **справа свободно** > **вправо** ПОКА < **снизу свободно** > **вниз** ПОКА < **слева свободно** > **влево** ПОКА < **сверху свободно** > **вверх** КОНЕЦ	B4

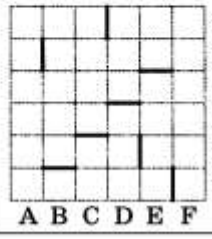
б)

Лабиринт	Программа	Начальная клетка
	НАЧАЛО ПОКА < **слева свободно** > **влево** ПОКА < **сверху свободно** > **вверх** ПОКА < **справа свободно** > **вправо** ПОКА < **снизу свободно** > **вниз** КОНЕЦ	F1

в)

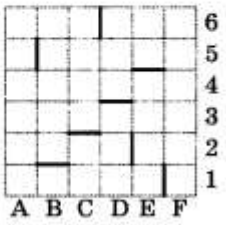
Лабиринт	Программа	Начальная клетка
	НАЧАЛО ПОКА < **слева свободно** > **влево** ПОКА < **сверху свободно** > **вверх** ПОКА < **справа свободно** > **вправо** ПОКА < **снизу свободно** > **вниз** КОНЕЦ	D3

г)

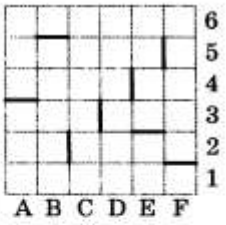
Лабиринт	Программа	Начальная клетка
	НАЧАЛО ПОКА < **слева свободно** > **влево** ПОКА < **справа свободно** > **вправо** ПОКА < **снизу свободно** > **вниз** ПОКА < **сверху свободно** > **вверх** КОНЕЦ	C5

5.4. Для исполнителя РОБОТ, описанного в предыдущем задании составить программу не более чем из семи строк (включая строки «НАЧАЛО» и «КОНЕЦ»), обеспечивающую нужное перемещение по заданному лабиринту

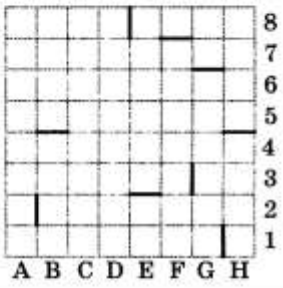
а)

Лабиринт	Программа	Начальная клетка	Конечная клетка
		A1	F6

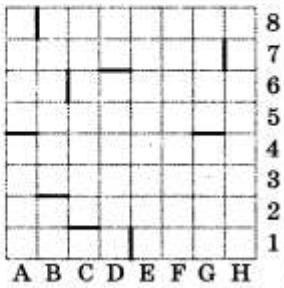
б)

Лабиринт	Программа	Начальная клетка	Конечная клетка
		C2	E5

в)

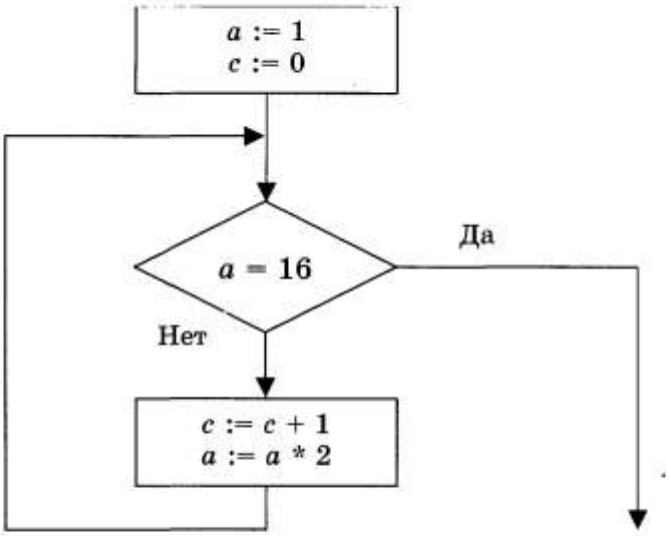
Лабиринт	Программа	Начальная клетка	Конечная клетка
		H1	D8

г)

Лабиринт	Программа	Начальная клетка	Конечная клетка
		D1	C2

Для решения задач на определение значения переменной после выполнения фрагмента программы или блок-схемы удобно использовать таблицу значений переменных.

Пример. Дана блок-схема алгоритма. Требуется найти значение переменной **c** после завершения алгоритма.



Решение

1-й способ

Составим таблицу значений переменных, добавив в нее для удобства результаты вычисления логического выражения.

№ шага	Значение **a**	Значение **c**	**a** = 16 ?
0	1	0	нет
1	1	1	
2	2	1	нет
3	2	2	
4	4	2	нет
5	4	3	
6	8	3	нет
7	8	4	
7	16	4	да

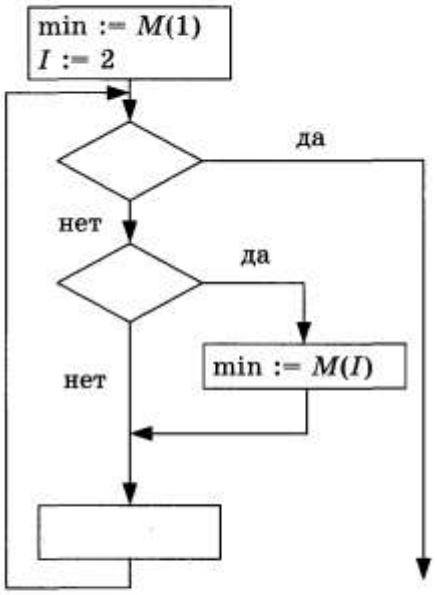
Итак, ответ **c** = 4.

2-й способ

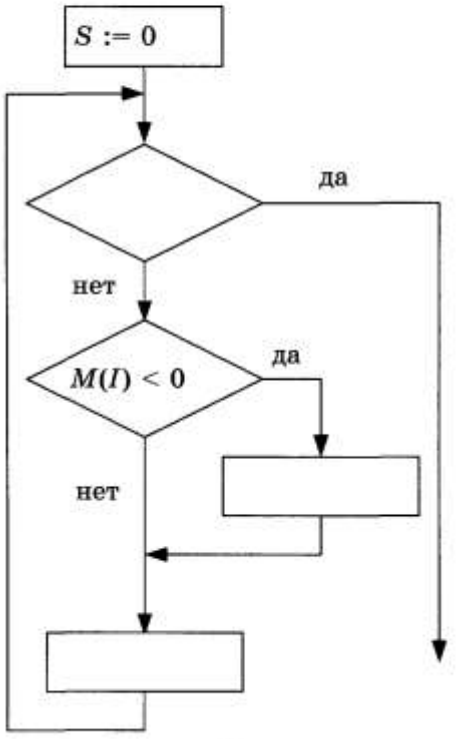
Внимательно проанализировав блок-схему можно заметить, что она иллюстрирует последовательное удвоение переменной **a**, изначально равной единице, до тех пор, пока она не достигнет значения $16 = 2^4$. Отсюда, $c = \log_2 16 = 4$.

5.5. Впишите во фрагмент блок-схемы пропущенные инструкции, необходимые для правильного решения поставленной задачи.

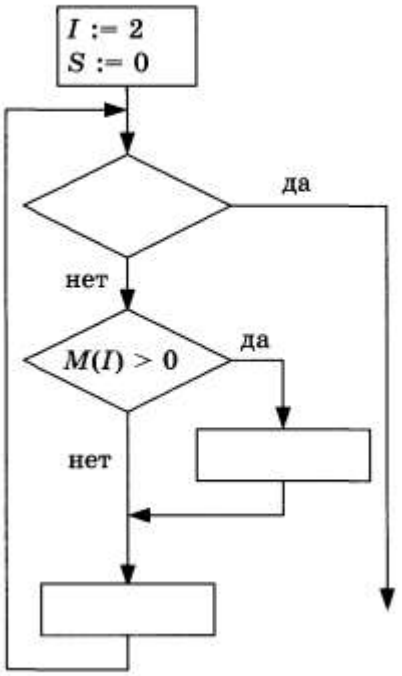
а) Найти минимальный элемент массива M из 24 элементов.



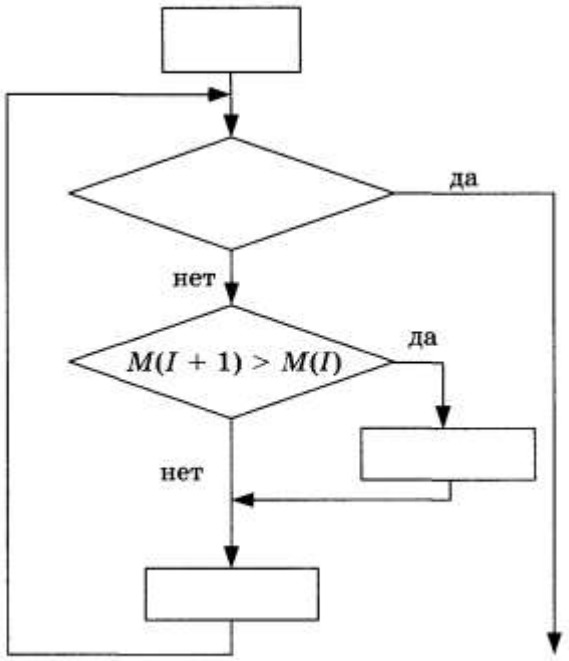
б) Найти сумму отрицательных элементов массива M из 24 элементов.



в) Найти сумму положительных элементов массива M из 22 элементов, имеющих четные номера (нумерация элементов массива начинается с единицы).

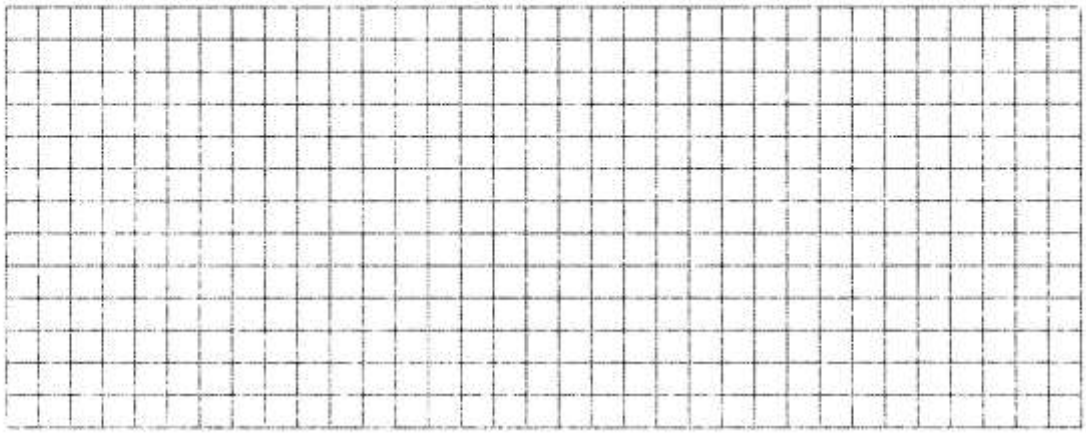


г) В массиве M 45 элементов. Найти, сколько подряд идущих элементов этого массива, начиная с первого, образуют возрастающую последовательность.

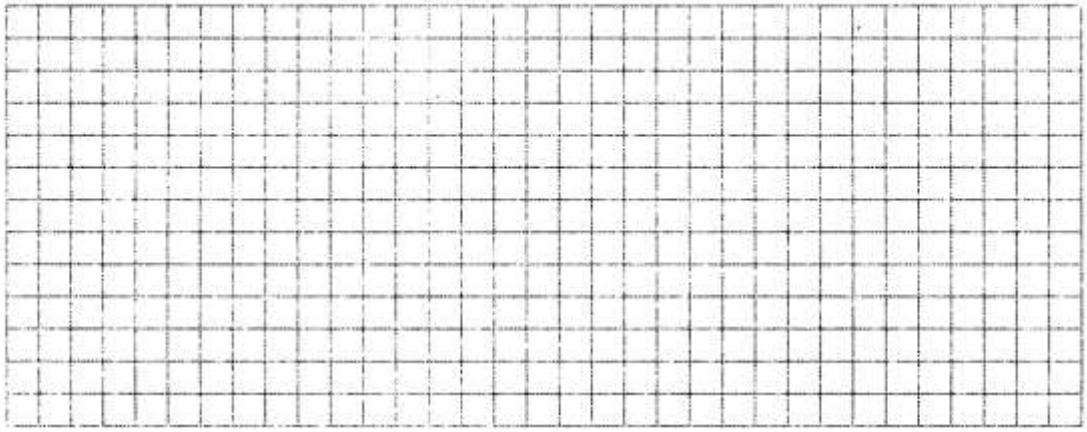


5.6. Составить и записать на естественном языке алгоритм, выполняющий:

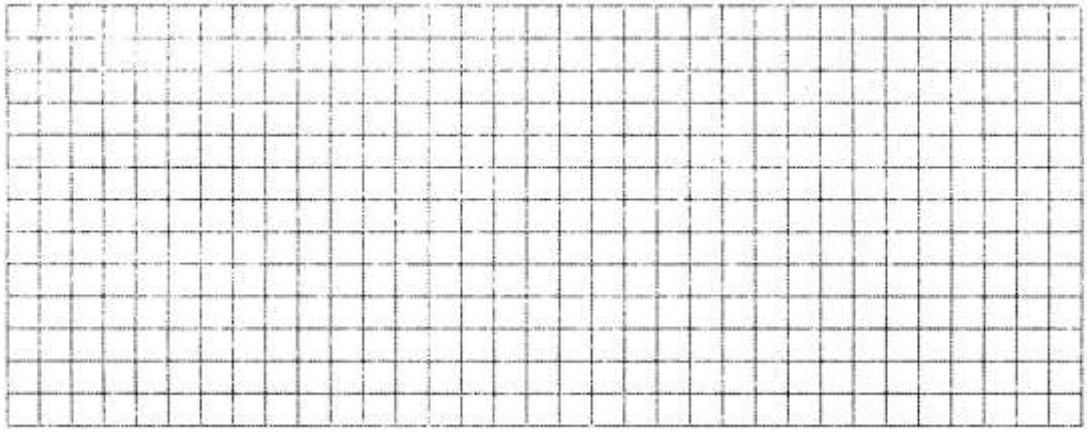
а) решение уравнения $ax = b$, где a и b — числовые исходные данные;



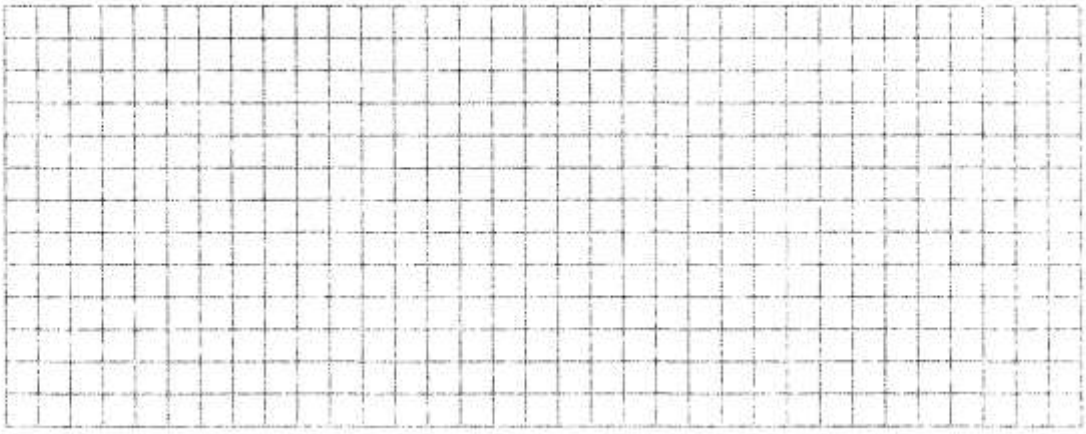
б) поиск количества минимальных элементов в заданном целочисленном массиве из 50 элементов;



в) поиск наибольшего общего делителя двух натуральных чисел;

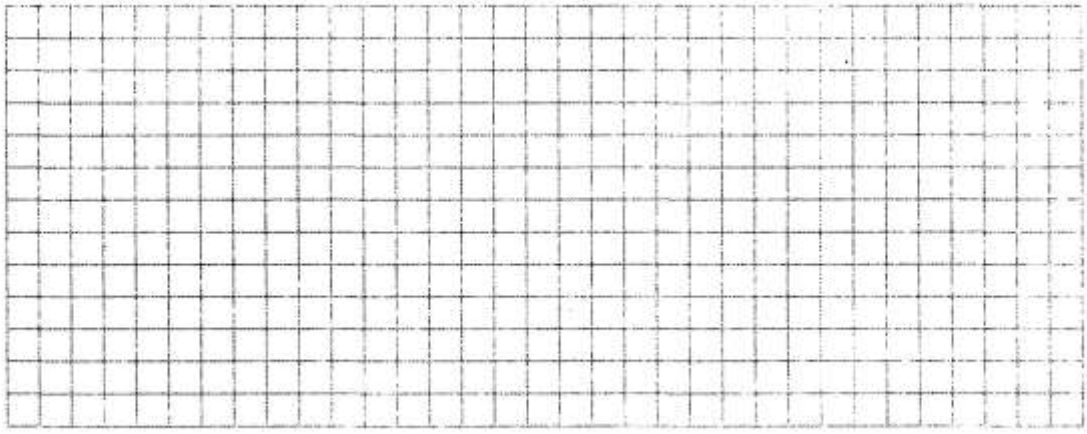


г) поиск количества простых чисел в заданном целочисленном массиве из 50 элементов.

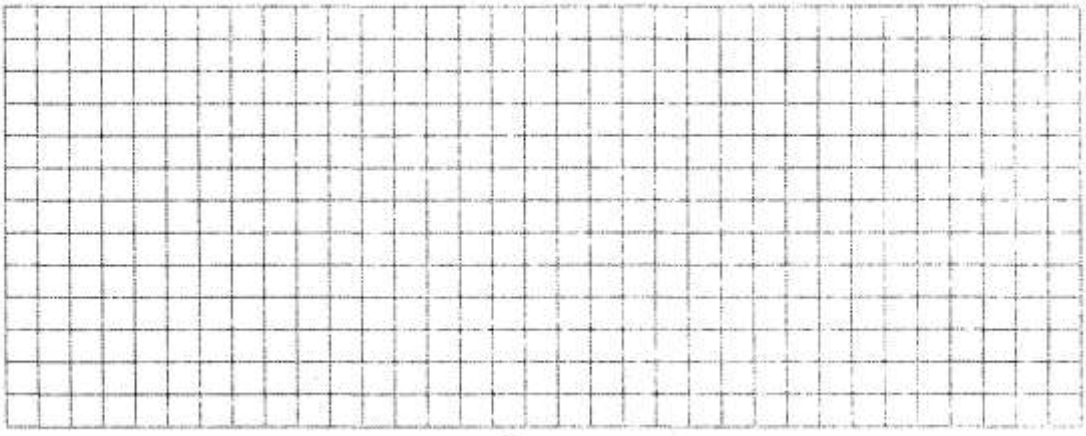


5.7. Составить и записать в виде блок-схемы алгоритм, выполняющий:

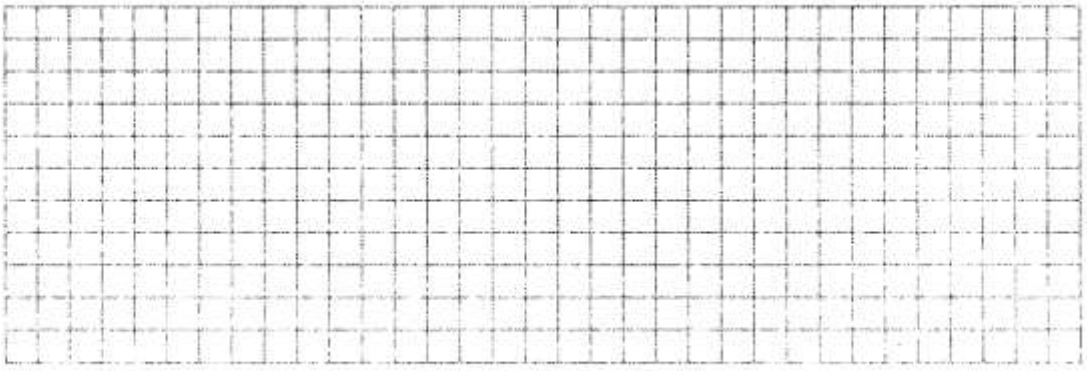
а) решение уравнения $ax = b$, где a и b — числовые исходные данные;



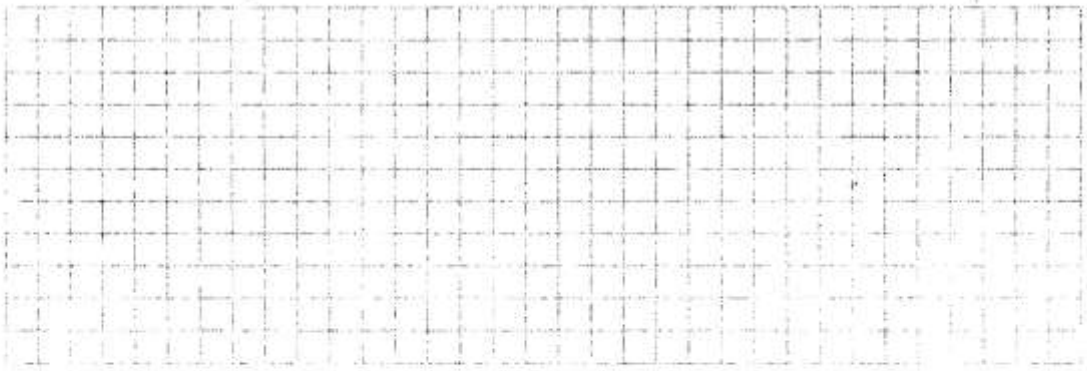
б) поиск количества минимальных элементов в заданном целочисленном массиве из 50 элементов;



в) поиск наибольшего общего делителя двух натуральных чисел;



г) поиск количества простых чисел в заданном целочисленном массиве из 50 элементов.



В задачах на определение результатов выполнения программы для одномерных и двумерных массивов необходимо определить закономерность обработки массива, поскольку его размерность может быть задана переменной величиной, а не конкретным значением.
Однако для того, чтобы понять эту закономерность, иногда требуется выполнить несколько итераций цикла «вручную», используя табличное представление массива.

Пример: Значения двумерного массива размера $n \times n$ задаются с помощью вложенного оператора цикла в представленном фрагменте программы.

Бейсик	Паскаль	Алгоритмический

FOR i=1 TO n
FOR k=1 TO n
IF i=k THEN
A(i,k) = 2
ELSE
A(i,k) = 0
END IF
NEXT k
NEXT n
	for i:=1 to n do
	for k:=1 to n do
	if i=k then
	A[i,k]:=2
	else
	A[i,k]:=0
		нц для i от 1 до n
		нц для k от 1 до n
		если i=k то
		A[i,k]:=2
		иначе
		A[i,k]:=0
		все
		кц
		кц

Как будет зависеть от n сумма элементов массива A после выполнения алгоритма?

Решение

Изобразим двумерный массив в виде таблицы. На диагонали таблицы, там, где $k = i$, будут находиться двойки. В остальных ячейках — нули.

i \ k	1	2	3	...	$n-1$	n
1	2	0	0	0	0	0
2	0	2	0	0	0	0
3	0	0	2	0	0	0
...	0	0	0	2	0	0
$n-1$	0	0	0	0	2	0
n	0	0	0	0	0	2

Количество элементов диагонали — n, поэтому сумма элементов массива будет равна $2n$.

Ответ: $2n$.

5.8. Выполнить фрагмент программы и найти итоговые значения переменных:

а)

Бейсик	Паскаль	Алгоритмический

a=1234
b=(a MOD 1000) · 10
a=a\1000+b
'\ и MOD — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно
	a:=1234;
	b:=(a mod 1000) · 10;
	a:=a div 1000+b;
	{div и mod — операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно}
		a:=1234
		b:=mod(a, 1000) · 10
		a:=div(a, 1000)+b
		div и mod — функции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно

$a =$ _____

$b =$ _____

б)

Бейсик	Паскаль	Алгоритмический

a=1234
b=(a MOD 10) · 1000
a=a\10+b
	a:=1234;
	b:=(a mod 10) · 1000;
	a:=a div 10+b;
		a:=1234
		b:=mod(a, 10) · 1000
		a:=div(a, 10)+b

$a =$ _____

$b =$ _____

в)

Бейсик	Паскаль	Алгоритмический
a=(7-5) · 4 b=(a MOD 3)+15 a=(b\4)+3	a:=(7-5) · 4; b:=(a mod 3)-15; a:=(b div 4)-3	a:=(7-5) · 4 b:=mod(a,3)+15 a:=div(b, 4)+3

$a =$ _____

$b =$ _____

г)

Бейсик	Паскаль	Алгоритмический

a=(6+2) * 4
b=(a MOD 5)+1
a=(b\6)-2
	a:=(6+2) * 4;
	b:=(a mod 5)+1;
	a:=(b div 6)-2
		a:=(6+2) * 4
		b:=mod(a,5)+1
		a:=div(b,6)-2

a = _____

b = _____

5.9.

а) Дан фрагмент программы, обрабатывающей массив *A* из 10 элементов:

Бейсик	Паскаль	Алгоритмический

n=10
FOR i = 1 TO n
A(i)=i
NEXT I
j = 1
FOR i = 1 TO n-1
IF A(i) <A(i+1) THEN j=j+1
NEXT I
	n=10;
	for i:=1 to n do A[i]:=i;
	j:=1;
	for i:=1 to n-1 do
	if A[i]<A[i+1]
	then j:=j+1;
		n:=10
		нц для i от 1 до n
		A[i]= i
		кц
		j:=1
		нц для i от 1 до n-1
		если A[i]<A[i+1] то
		j=j+1
		все
		кц

Чему будет равно значение переменной *j* после выполнения данного алгоритма?

Ответ: _____.

б) Дан фрагмент программы, обрабатывающей массив *A* из 10 элементов:

Бейсик	Паскаль	Алгоритмический

n=10
FOR i = 1 TO n
A(i)=11-i
NEXT I
s = 0
FOR i = 1 TO n
IF i>5 THEN s=s+A(i)
NEXT I
	n=10;
	for i:=1 to n do
	A[i]:=11-i;
	s:=0;
	for i:=1 to n do
	if i>5 then s:=s+A[i];
		n:=10
		нц для i от 1 до n
		A[i]= 11-i
		кц
		s:=0
		нц для i от 1 до n
		если i>5 то
		s=s+A[i]
		все
		кц

Чему будет равно значение переменной *s* после выполнения данного алгоритма?

Ответ: _____.

в) Дан фрагмент программы, обрабатывающей массив A из n элементов:

Бейсик	Паскаль	Алгоритмический

j = A(1)
FOR i = 2 TO n
IF j <A(i) THEN j=A(i)
NEXT I
	j:=A[1];
	for i:=2 to n do
	if j<A[i]
	then j:=A[i];
		j:=A[1]
		нц для i от 2 до n
		если j<A[i] то
		j:=A[i]
		все
		кц

Как зависит итоговое значение переменной j от значений массива A?

Ответ: _____.

г) Дан фрагмент программы, обрабатывающей массив A из n элементов:

Бейсик	Паскаль	Алгоритмический

j = 0
FOR i = 1 TO n
IF A(i)<0 THEN j=j+1
NEXT I
	j:=0;
	for i:=1 to n do
	if A[i]<0
	then j:=j+1;
		j:=0
		нц для i от 1 до n
		если A[i]<0 то
		j:=j+1
		все
		кц

Как зависит итоговое значение переменной j от значений массива A?

Ответ: _____.

5.10.

а) Значения двумерного массива размера $n \times n$ задаются с помощью вложенного оператора цикла в представленном фрагменте программы:

Бейсик	Паскаль	Алгоритмический

FOR i=1 TO n
FOR k=1 TO n
IF i>k THEN
A(i,k) = 1
ELSE
A(i,k) = 0
END IF
NEXT k
NEXT n
	for i:=1 to n do
	for k:=1 to n do
	if i>k then
	A[i,k]:=1
	else
	A[i,k]:=0
		нц для i от 1 до n
		нц для k от 1 до n
		если i>k то
		A[i,k]:=1
		иначе
		A[i,k]:=0
		все
		кц
		кц

Как будет зависеть от n сумма элементов массива A после выполнения алгоритма? Напишите формулу вычисления суммы элементов массива A, в зависимости от n.

Ответ: _____.

б) Значения двумерного массива размера $n \times n$ задаются с помощью вложенного оператора цикла в представленном фрагменте программы:

Бейсик	Паскаль	Алгоритмический

FOR i=1 TO n
FOR k=1 TO n
IF i=k THEN
  A(i,k) = 1
ELSE
  A(i,k) = -1
END IF
NEXT k
NEXT n
	for i:=1 to n do
	for k:=1 to n do
	if i=k then
	  A[i,k]:=1
	else
	  A[i,k]:=-1
		нц для i от 1 до n
		нц для k от 1 до n
		  если i=k то
		  A[i,k]:=1
		  иначе
		  A[i,k]:=-1
		  все
		кц
		кц

Как будет зависеть от n сумма элементов массива A после выполнения алгоритма? Напишите формулу вычисления суммы элементов массива A, в зависимости от n.

Ответ: _____.

в) Значения двумерного массива размера $n \times n$ задаются с помощью вложенного оператора цикла в представленном фрагменте программы:

Бейсик	Паскаль	Алгоритмический

FOR i=1 TO n
FOR k=1 TO n
IF i>k THEN
  A(i,k) = 1
ELSE
  A(i,k) = -1
END IF
NEXT k
NEXT n
	for i:=1 to n do
	for k:=1 to n do
	if i>k then
	  A[i,k]:=1
	else
	  A[i,k]:=-1
		нц для i от 1 до n
		нц для k от 1 до n
		  если i>k то
		  A[i,k]:=1
		  иначе
		  A[i,k]:=-1
		  все
		кц
		кц

Как будет зависеть от n сумма элементов массива A после выполнения алгоритма? Напишите формулу вычисления суммы элементов массива A, в зависимости от n.

Ответ: _____.

г) Значения двумерного массива размера $n \times n$ задаются с помощью вложенного оператора цикла в представленном фрагменте программы:

Бейсик	Паскаль	Алгоритмический

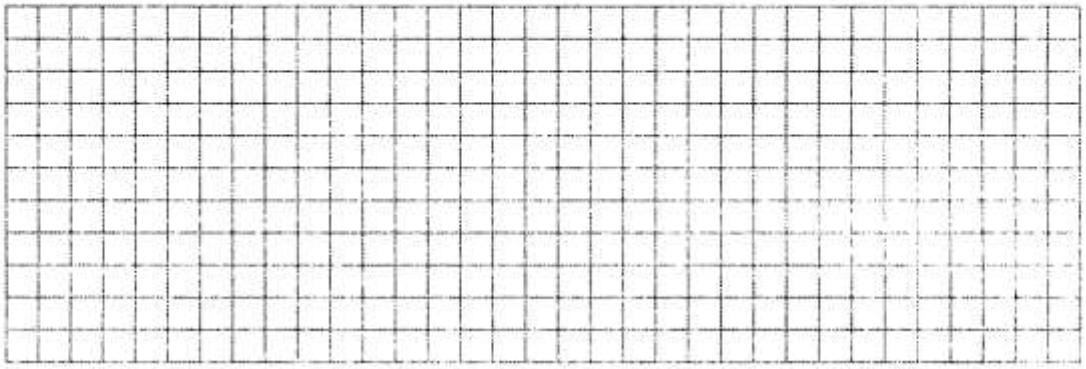
FOR i=1 TO n
FOR k=1 TO n
IF i>k THEN
  A(i,k) = i
ELSE
  A(i,k) = -k
END IF
NEXT k
NEXT n
	for i:=1 to n do
	for k:=1 to n do
	if i>k then
	  A[i,k]:=i
	else
	  A[i,k]:=-k
		нц для i от 1 до n
		нц для k от 1 до n
		  если i>k то
		  A[i,k]:=i
		  иначе
		  A[i,k]:=-k
		  все
		кц
		кц

Как будет зависеть от n сумма элементов массива A после выполнения алгоритма? Напишите формулу вычисления суммы элементов массива A, в зависимости от n.

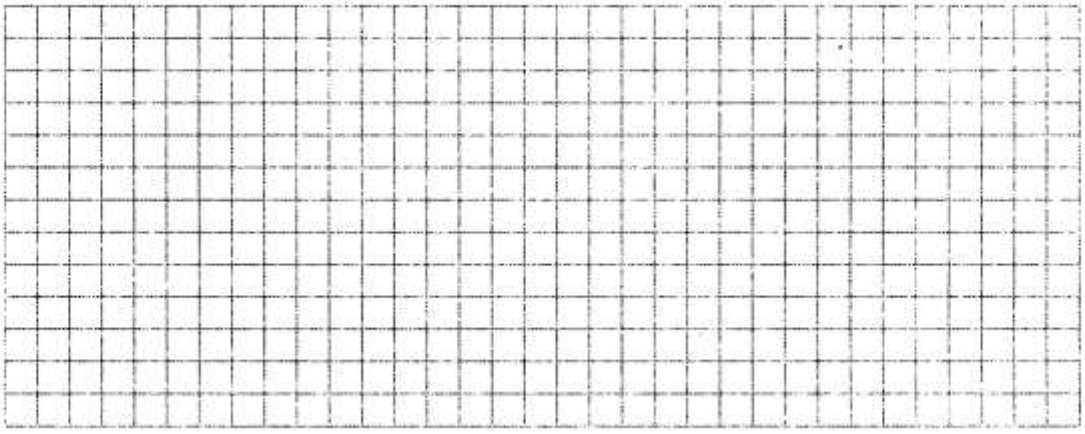
Ответ: _____.

5.11. Составить и записать в виде программы на любом языке программирования алгоритм, выполняющий:

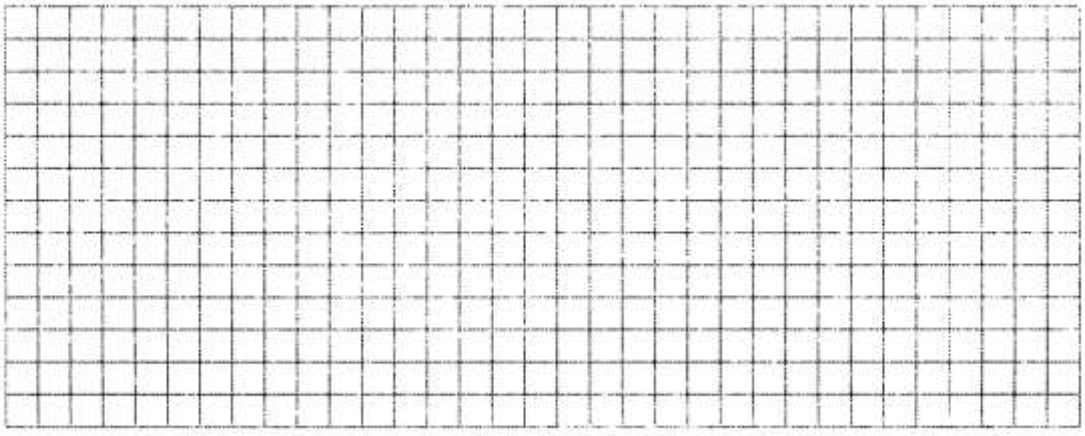
а) Решение уравнения $a|x| = b$, где a и b — исходные числовые данные;



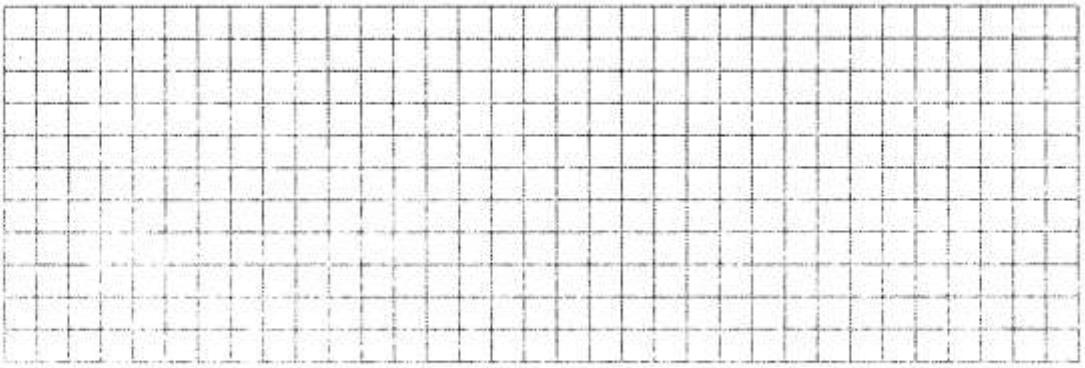
б) Поиск количества минимальных элементов в заданном целочисленном массиве из 50 элементов;



в) Поиск наибольшего общего делителя двух натуральных чисел;



г) Поиск количества простых чисел в заданном целочисленном массиве из 50 элементов;



Важные рекомендации

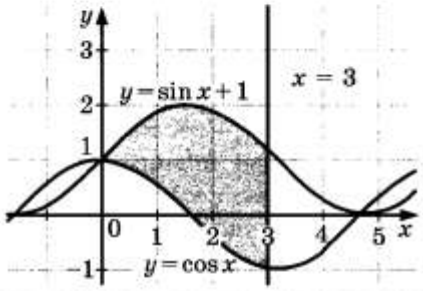
Практика показывает, что наибольшие затруднения при выполнении заданий по теме «Алгоритмизация и программирование» вызывает составление программ и алгоритмов (задачи С1, С2 и С4 демоверсии ЕГЭ 2009 г.). Данное пособие не является учебником программирования, поэтому мы приведем лишь общие рекомендации, не зависящие от языка, на котором записываются программы:

- Перед составлением программы надо тщательно проанализировать условие задания, уяснить постановку задачи, формат входных и выходных данных.
- Приступать к написанию программы можно лишь тогда, когда детально продуман алгоритм решения задачи, не следует начинать писать программу, не представляя алгоритмического решения. Для простой задачи алгоритмическое решение можно держать и в голове, но лучше его зафиксировать на бумаге в виде пояснения или схемы.
- Для написания программы нужно пользоваться тем языком программирования, которым хорошо владеешь и который подходит для решения поставленной задачи. Не следует специально стремиться поразить экзаменаторов знанием экзотического или недавно появившегося языка программирования, поскольку дополнительные баллы за это не предусмотрены. Хорошая программа должна быть максимально простой и эффективной. Не нужно специально использовать сложные приемы или редко применяемые конструкции языка программирования, если в этом нет необходимости.
- Текст программы нужно писать аккуратно и разборчиво даже на черновике, структурировано, выделяя отступами уровень вложенности операторов, не скупясь на комментарии и пояснения. Отсутствие комментариев и пояснений вовсе не является признаком высокой квалификации программиста, скорее наоборот.
- При написании текста программы следует контролировать соответствие типов переменных и применяемых к ним операций, не допускать использования неопределенных (неинициализированных) значений переменных.
- Если текст программы получается, на ваш взгляд, очень сложным и запутанным, приостановите на время ее написание и подумайте, в чем причина этой сложности, нельзя ли найти более простое и изящное решение задачи.
- После того как программа написана, необходимо выполнить ее тестирование, то есть проверку правильности работы для различных исходных данных. Постарайтесь найти ошибку в своей программе, помните, что цель тестирования — поиск ошибок, а не демонстрация правильности программы. Гораздо лучше, если ошибку найдете и исправите вы сами, а не экзаменаторы при проверке.

- При тестировании следует подбирать набор тестовых данных так, чтобы все операторы, все «ветки» программы выполнились на совокупности тестов. Особое внимание следует уделить особым случаям исходных данных, пограничным, критическим точкам. Пусть, например, написана программа сортировки массива. Особыми здесь будут случаи, когда входной массив уже отсортирован в нужном порядке, когда он отсортирован в порядке обратном требуемому, а также когда все элементы сортируемого массива равны. Следует выявить такие критические точки и убедиться, что программа правильно работает на этих тестах.
- Если в задании приведен пример входных данных, его тоже надо включить в состав тестов, но он не должен быть единственным тестом!
- При тестировании следует еще раз сверить структуру исходных и выходных данных с заданием, убедиться, что составленная программа точно соответствует условию задачи.
- При тестировании необходимо внимательно контролировать синтаксис программы (должен быть явно указан тип каждой переменной, если этого требует язык программирования, переменные должны использоваться в соответствии со своим типом, не должно быть неинициализированных и неиспользуемых переменных, должны быть правильно расставлены операторные скобки и разделители операторов и т.д.)
- Если после тестирования программы осталось достаточно времени, то можно поработать над повышением ее эффективности, подумать, нельзя ли упростить ее, уменьшить количество циклов или требуемой для ее работы памяти. В случае внесения изменений в готовую программу процедуру тестирования придется повторить заново.

5.12.

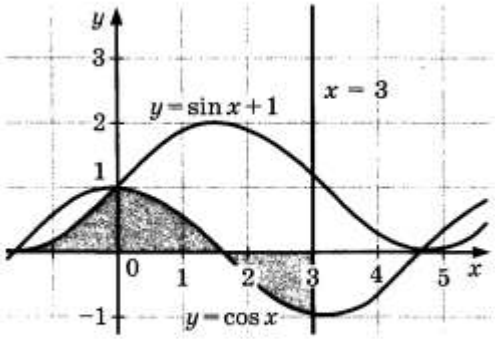
а) Найдите и исправьте ошибки в приведенной программе проверки принадлежности точки закрашенной области. Приведите пример, когда исходная программа работала неправильно.



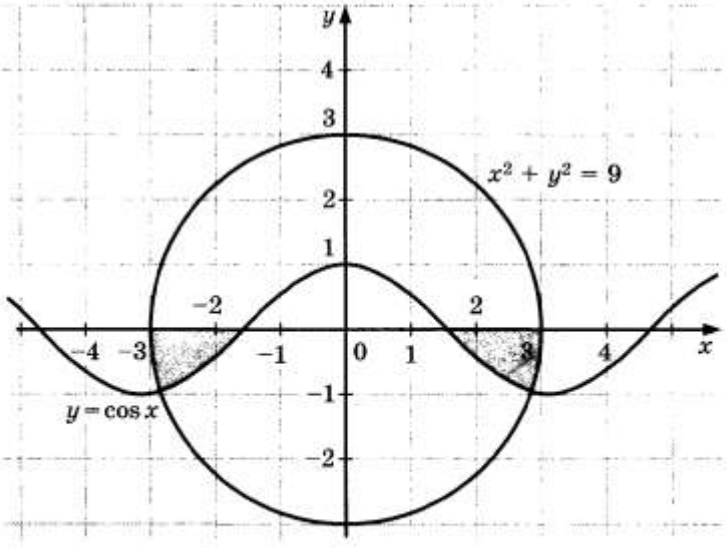
Программа на Паскале	Программа на Бейсике	Программа на Си

var x,y: real;
begin
readln(x,y);
if y<=sin (x)+1 then
if y>= cos (x) then
if x<=3 then
write('принадлежит')
else
write('не принадлежит')
end.
	INPUT x, y
	IF y<=sin(x)+1 THEN
	IF y>= cos(x) THEN
	IF x<=3 THEN
	PRINT "принадлежит"
	ELSE
	PRINT "не принадлежит"
	ENDIF
	ENDIF
	ENDIF
	END
		void main(void)
		{ float x,y;
		scanf("%f%f",&x,&y);
		if (y<=sin(x)+1)
		if (y>=cos(x))
		if (x<=3)
		printf("принадлежит");
		else
		printf("не принадлежит");
		}

б) Составьте программу проверки принадлежности точки закрашенной области.



в) Найдите и исправьте ошибки в приведенной программе проверки принадлежности точки закрашенной области. Приведите пример, когда исходная программа работала неправильно.

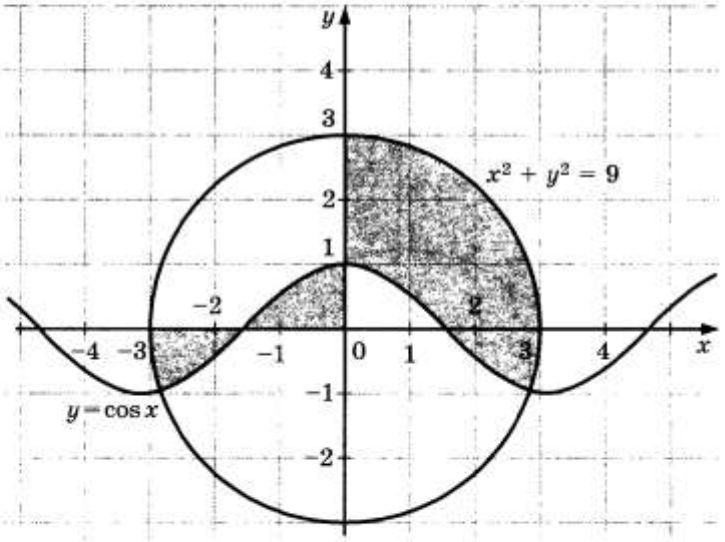


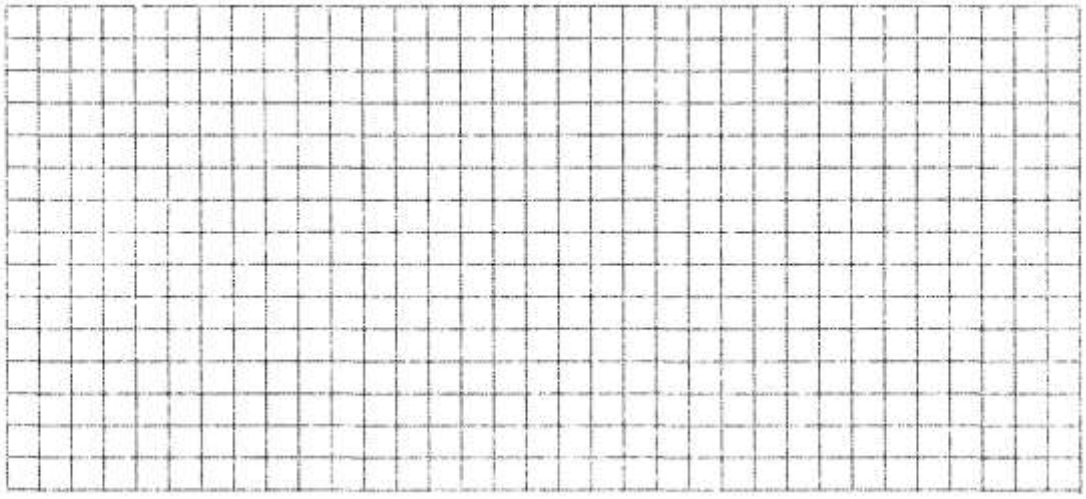
Программа на Паскале	`var x,y: real;`
	`begin`
	`  readln(x,y);`
	`  if (sqr(x) + sqr(y) <=3 ) or (y>=cos(x)) or (y<=0))then`
	`  write('принадлежит')`
	`  else`
	`  write('не принадлежит')`
	`end.`

Программа на Бейсике	`INPUT x, y`
	`IF (x * x+ y * y <= 3) or (y>=cos(x) )or (y<=0) THEN`
	`  PRINT "принадлежит"`
	`  ELSE`
	`  PRINT "не принадлежит"`
	`ENDIF`
	`END`

Программа на Си	`void main(void)`
	`{ float x,y;`
	`  scanf("%f%f",&x,&y);`
	`  if ((x * x+ y * y <= 3) || (y>=cos(x) )|| (y<=0))`
	`  printf("принадлежит");`
	`  else`
	`  printf("не принадлежит");`
	`}`

Составьте программу проверки принадлежности точки закрашенной области.





5.13.

а) Составить программу, выводящую на экран все натуральные трехзначные числа, делящиеся без остатка на 17, сумма цифр которых равняется 11.