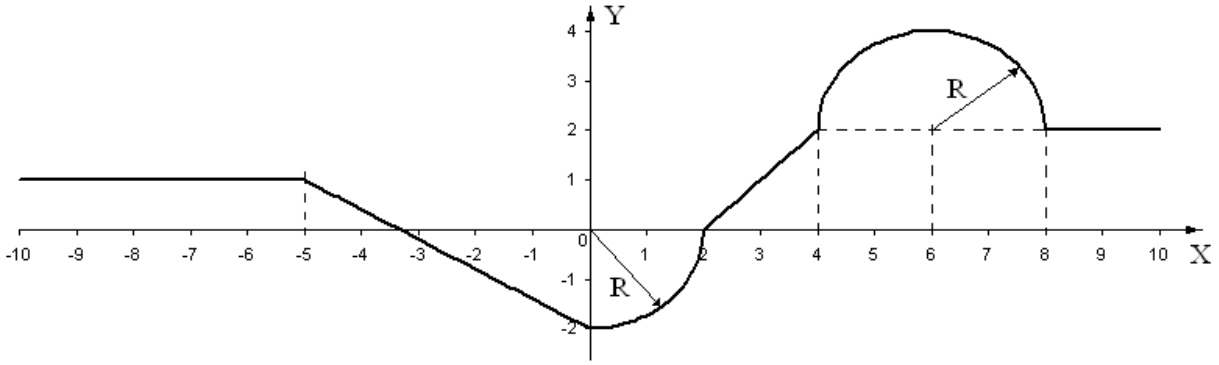
Лабораторная работа №2: «Разветвляющиеся вычислительные процессы», Задание 1

Цель работы:

Дать студентам практический навык в использовании условных операторов ветвления на языке программирования Python. Работа состоит из двух заданий.

Постановка задачи

Написать программу, которая по введённому значению аргумента вычисляет значение функции, заданной в виде графика.



Теоретическая часть

Для решения задачи использован оператор ветвления, который в языке Python имеет следующий вид:

```
if <Логическое выражение>:
    <Блок – выполняется, если условие истинно>
[elif <Логическое выражение>:
    <Блок – выполняется, если условие истинно>
]
[else:
     <Блок – выполняется, если все условия ложны>
]
```

<Блок> – это набор вложенных инструкций, которые выделяются одинаковым количеством пробелов (обычно четырьмя).

Для ввода данных используется инструкция `input()`, которая возвращает строку. Введённые значения, перед использованием в арифметических выражениях, должны быть преобразованы к числовому формату.

Вывод данных выполняется инструкцией `print()`, в которой использован форматированный вывод данных.

График функции представлен фрагментами прямых линий, описываемых уравнением $y = kx + b$ и дугами кругов. В общем случае уравнение круга может быть представлено так: $(x-a)^2 + (y-b)^2 = R^2$.

Неизвестные параметры, угол наклона и смещение прямой, а так же координаты центра дуг, определим, используя данные из графика.

Для прямой на интервале (-5, 0) можем записать следующую систему уравнений:

$$\begin{cases} 1 = k \cdot (-5) + b \\ -2 = k \cdot 0 + b \end{cases}$$

Из второго уравнения следует, что b = -2, а из первого – k = -3/5.

Для полукруга с центром (6, 2) уравнение круга примет вид:

$$(x-6)^2+(y-2)^2=2^2$$

Перепишем уравнение так:

$$(y-2)^2=2^2-(x-6)^2$$

$(y-2)^2=4-(x-6)^2$ отсюда следует, что $y=2+\sqrt{4-(x-6)^2}$. Знак перед корнем выбран для случая, когда рассматривается верхняя часть полукруга.

Выполнив необходимые вычисления для всех фрагментов функции, мы получим систему уравнений, которую запишем в следующем виде:

$$y=\begin{cases} 1 & x<-5 \\ -\frac{3}{5}x-2 & -5<=x<0 \\ -\sqrt{4-x^2} & 0<=x<2 \\ x-2 & 2<=x<4 \\ 2+\sqrt{4-(x-6)^2} & 4<=x<8 \\ 2 & x>=8 \end{cases}$$

Функция определена на всём диапазоне $x\in(-\infty;+\infty)$. При этом, особых точек у неё нет.

Описание программы

Программа написана на алгоритмическом языке Python 3.6, реализована в среде ОС Windows 10 и состоит из частей, отвечающих за ввод данных, вычисление и представление данных на экране монитора.

Описание алгоритма

1. Ввести значение аргумента `x` и преобразовать его к типу `float`.
2. Определить, к какому интервалу из области определения функции оно принадлежит, и вычислить значение функции `y` по соответствующей формуле.
3. Вывести значение `x` и `y`.

Описание входных и выходных данных

Входные данные поступают с клавиатуры, а выходные - выводятся на монитор для просмотра. Входные и выходные данные имеют тип `float`.

Листинг программы (вариант 1)

```
# -*- coding: cp1251 -*-
from math import * # теперь можно так:
                   # print(sin(pi/4))
x = float(input('Введите значение x='))
if x < -5: y = 1
if x >=-5 and x<0: y = -(3/5)*x-2
if x >= 0 and x<2: y = -sqrt(4-x**2)
if x >= 2 and x<4: y = x-2
```

```
if x >= 4 and x<8: y = 2+sqrt(4-(x-6)**2)
if x >= 8: y = 2
print("X={0:.2f}    Y={1:.2f}".format(x, y))
```

Блок-схема алгоритма этого решения приведена в Приложении 1 (рис.1) к лабораторной работе.

Следует отметить, что в такой записи алгоритма проверка выполняется для всех условных операторов, в том числе и тех, которые следуют за вычисленным. Так, например, если x равно -3, то выполнится второй оператор, но и во всех последующих операторах операция сравнения будет проведена. Число проверок можно сократить, если написать программу с использованием вложенных условных операторов.

Листинг программы (вариант 2)

```
# -*- coding: cp1251 -*-
from math import * # теперь можно так:
                   # print sin(pi/4)
x = float(input('Введите значение x='))
if x < -5:
    y = 1
elif x >=-5 and x<0:
    y = -(3/5)*x-2
elif x >= 0 and x<2:
    y = -sqrt(4-x**2)
elif x >= 2 and x<4:
    y = x-2
elif x >= 4 and x<8:
    y = 2+sqrt(4-(x-6)**2)
else: y = 2
print("X={0:.2f}    Y={1:.2f}".format(x, y))
```

Блок-схема алгоритма решения приведена в Приложении 2 (рис.2) к лабораторной работе.

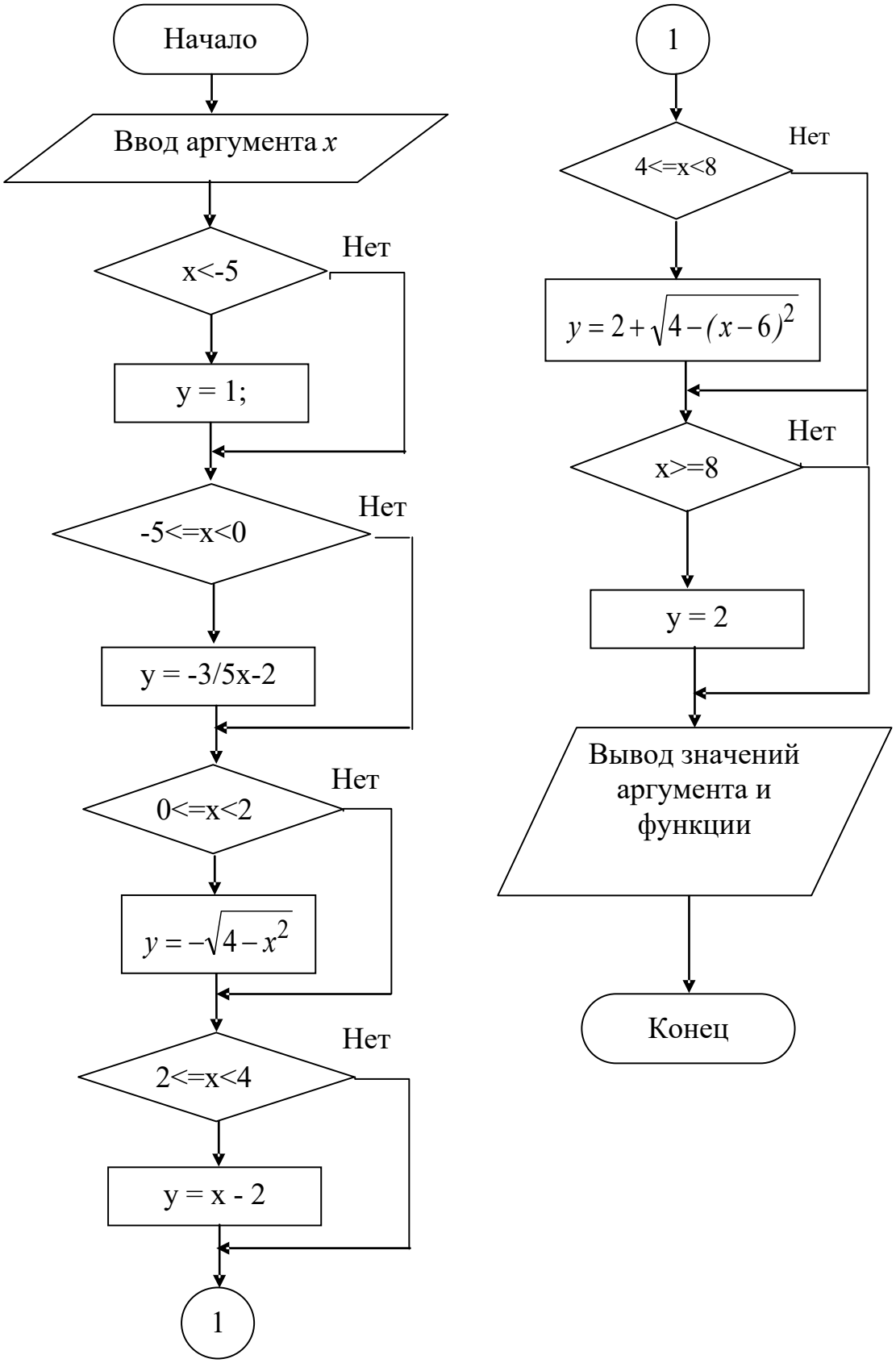
Результат работы программы

```
Введите значение аргумента: -6
X= -6.00      Y= 1
Введите значение аргумента: -3.33
X= -3.33      Y= -0.00
Введите значение аргумента: 6
X= 6.00      Y= 4.00
```

Список используемой литературы

1. Н.А. Прохоренок, В.А. Дронов, Python 3 и PyQt 5. Разработка приложений: СПб.: БХВ-Петербург, 2017

ПРИЛОЖЕНИЕ 1
к лабораторной работе №2, задание 1



к

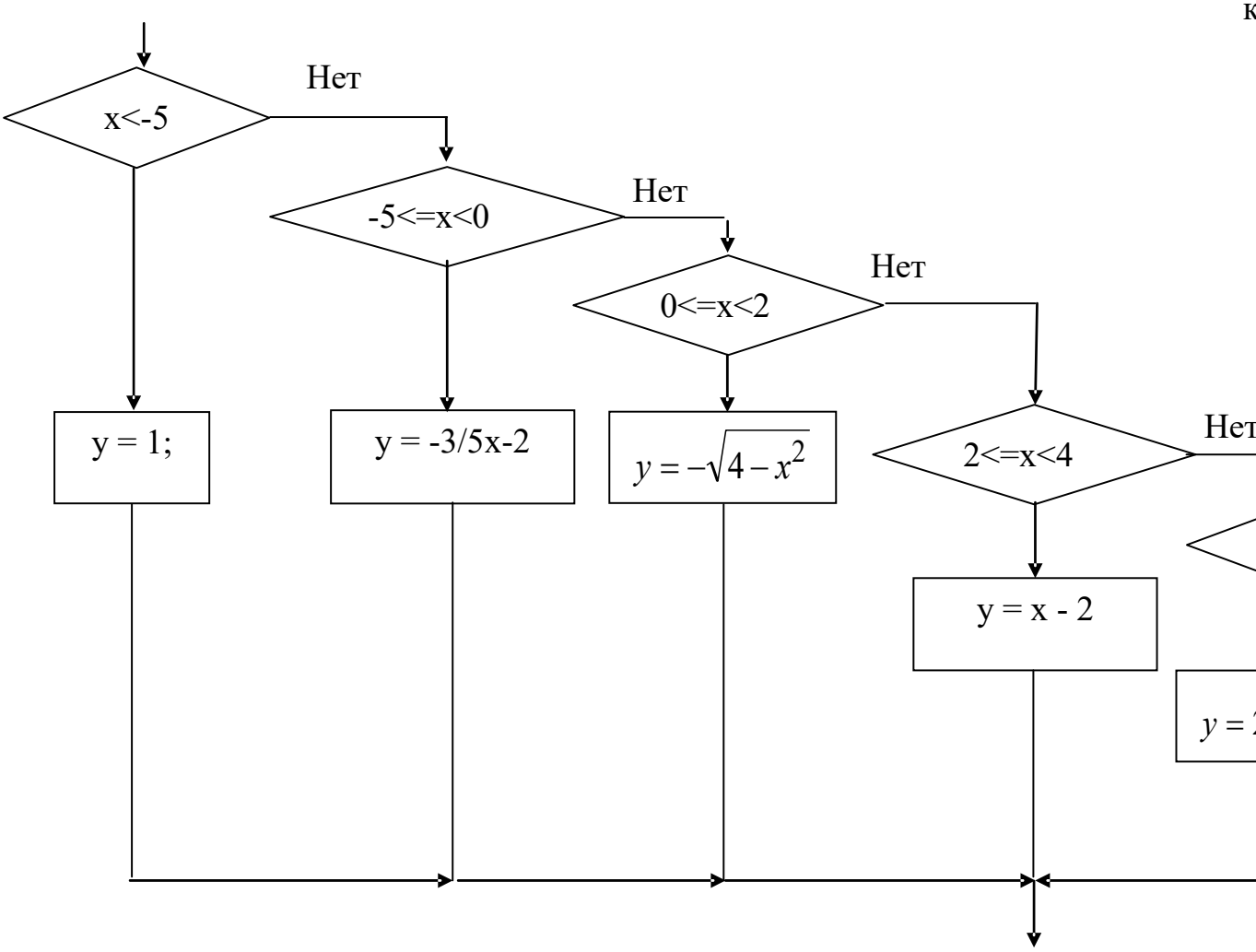


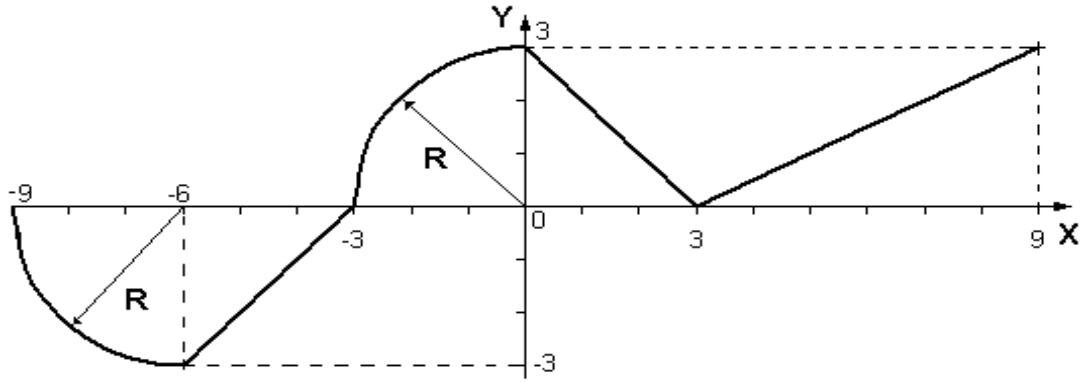
Рис.2 – Блок-схема алгоритма программы Lab2_1b (показана только логи

Задание к лабораторной работе №2 «Разветвляющиеся вычислительные процессы».

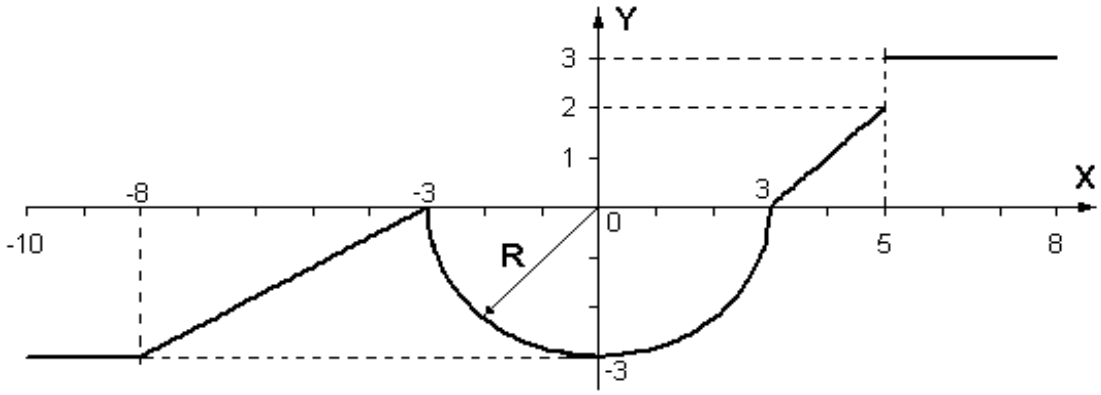
Задание 1

Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика. Параметры, необходимые для решения задания следует получить из графика и определить в программе.

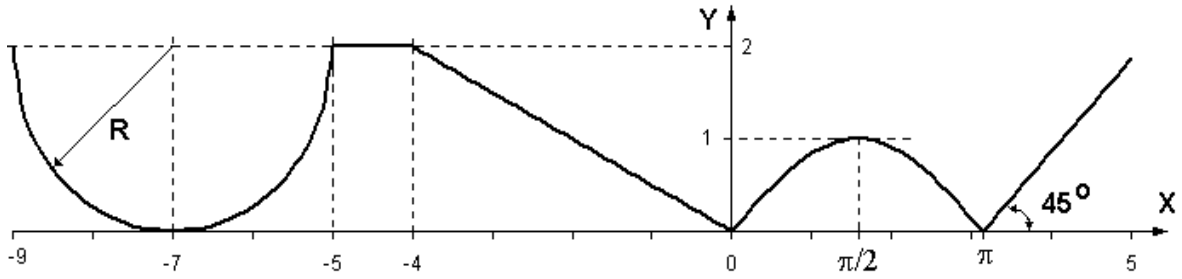
1)



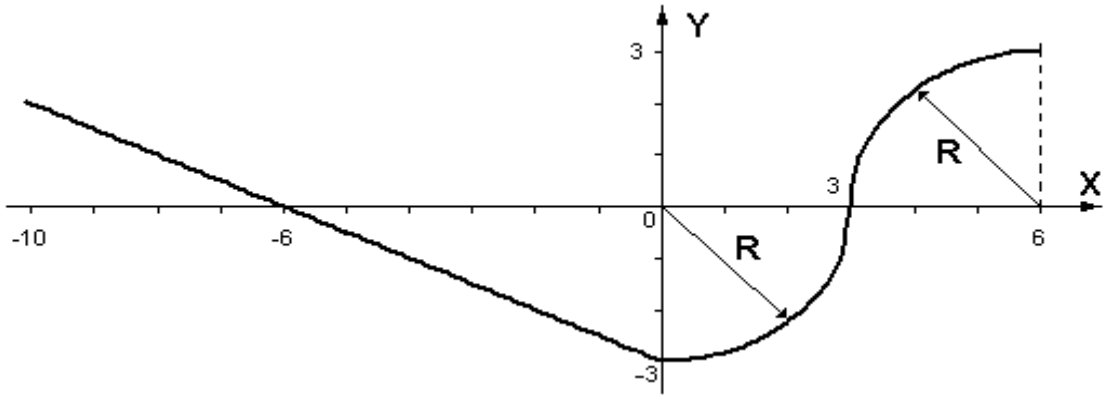
2)



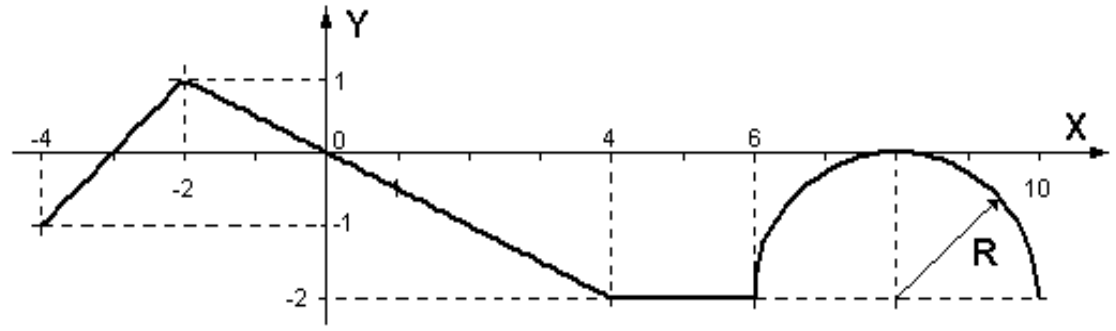
3)



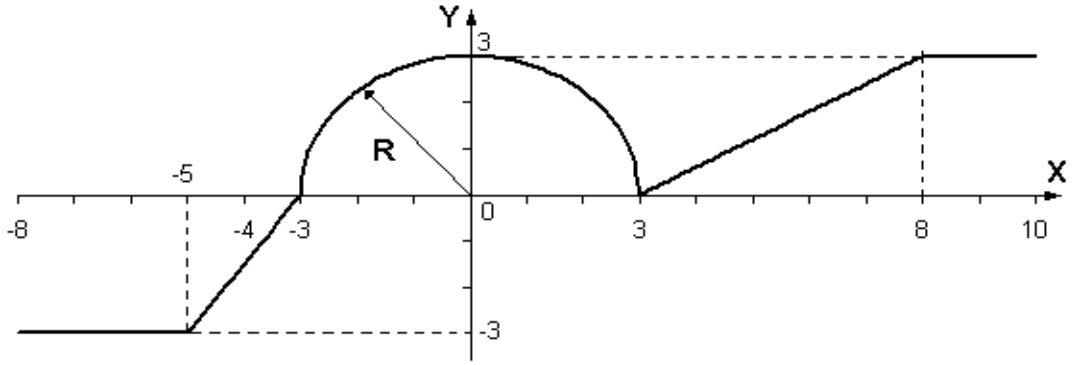
4)



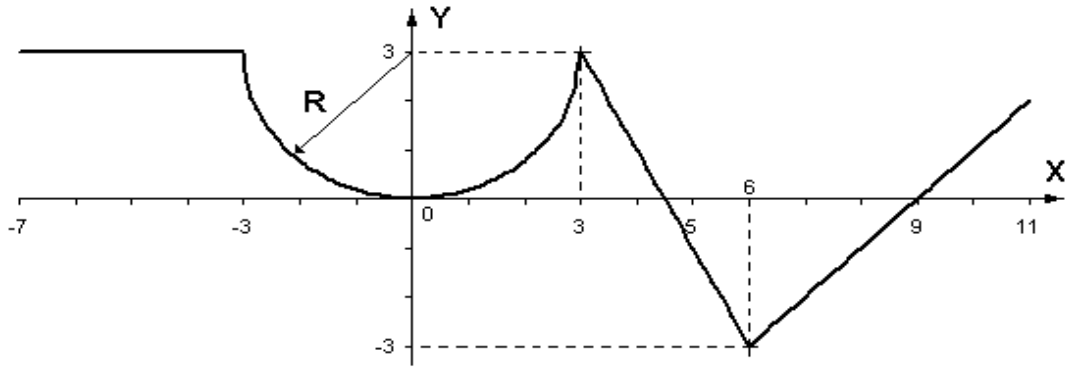
5)


Y
X
1
-1
-2
-4
-2
0
4
6
10
R

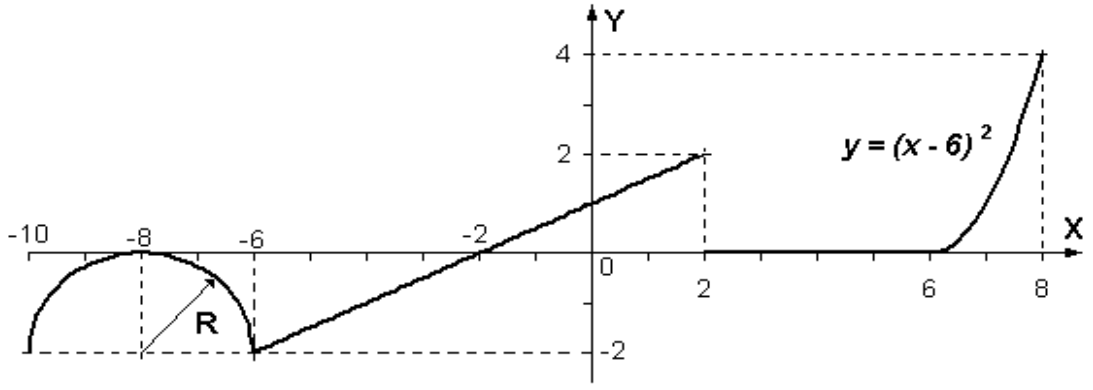
6)


Y
X
3
-3
-5
-8
-4
-3
0
3
8
10
R

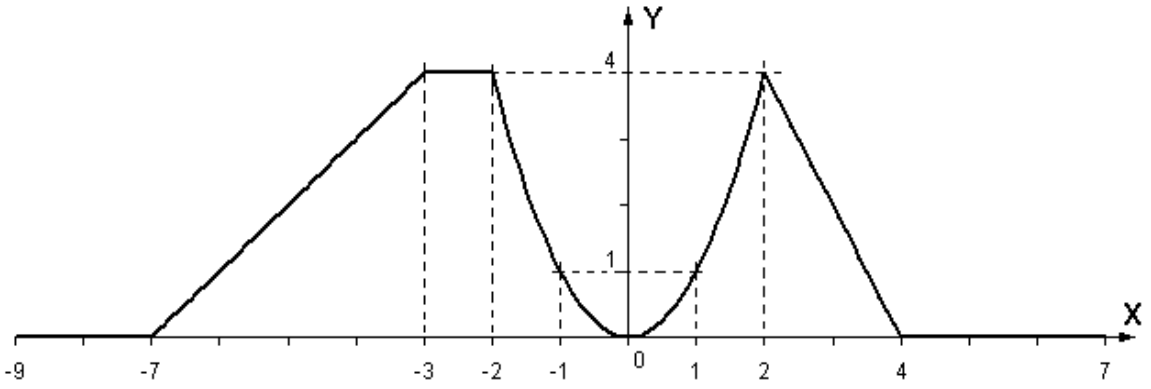
7)


Y
X
3
-3
R
-7
-3
0
3
5
6
9
11

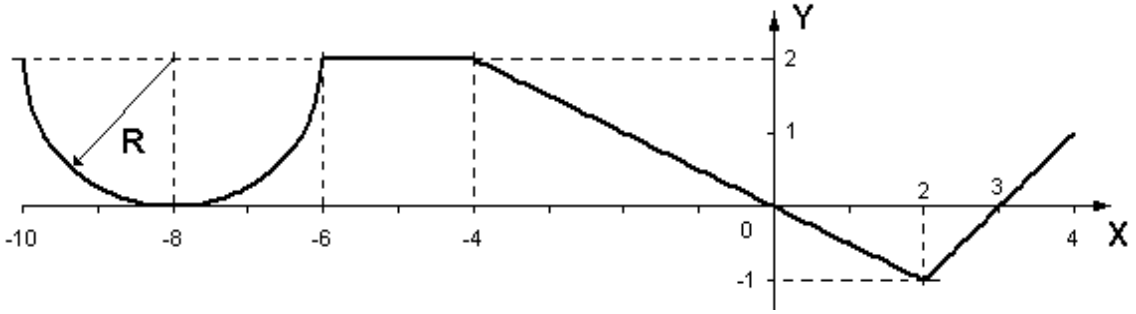
8)


Y
X
4
2
-2
y = (x - 6) ²
-10
-8
-6
-2
0
2
6
8
R

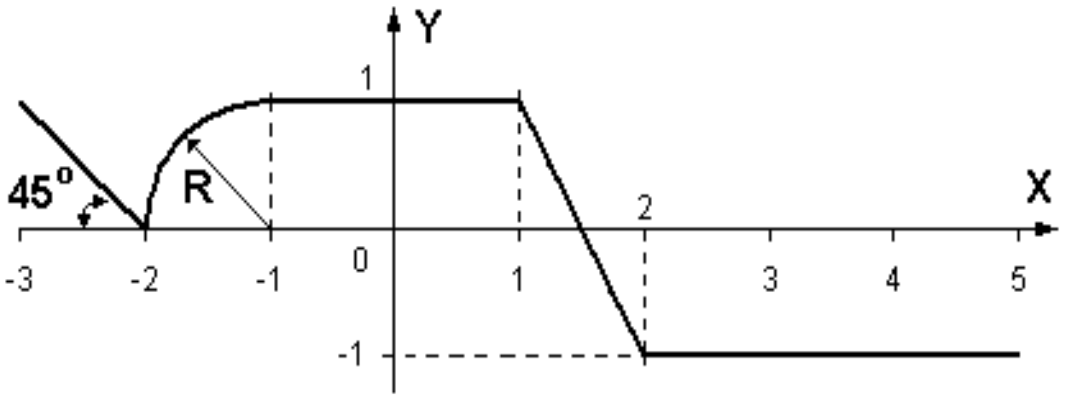
9)


Y
X
4
1
0
-9
-7
-3
-2
-1
1
2
4
7

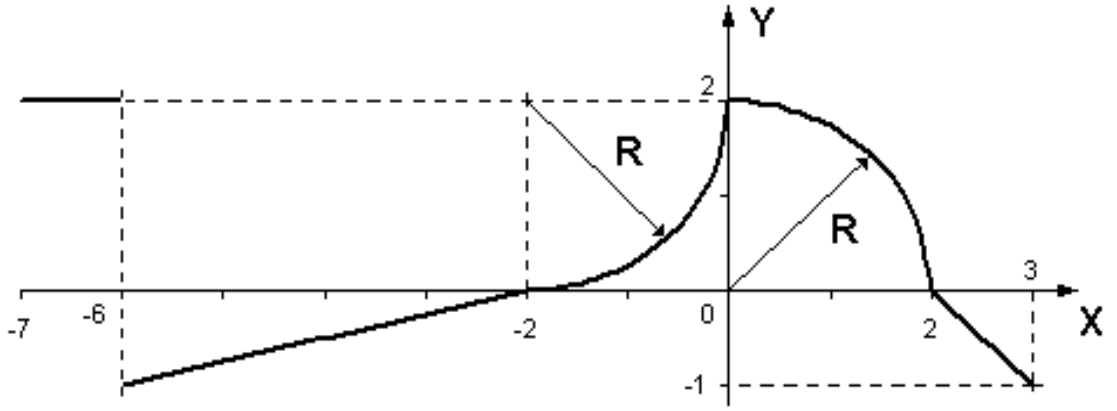
10)


Y
X
R
2
1
-1
-10
-8
-6
-4
0
2
3
4

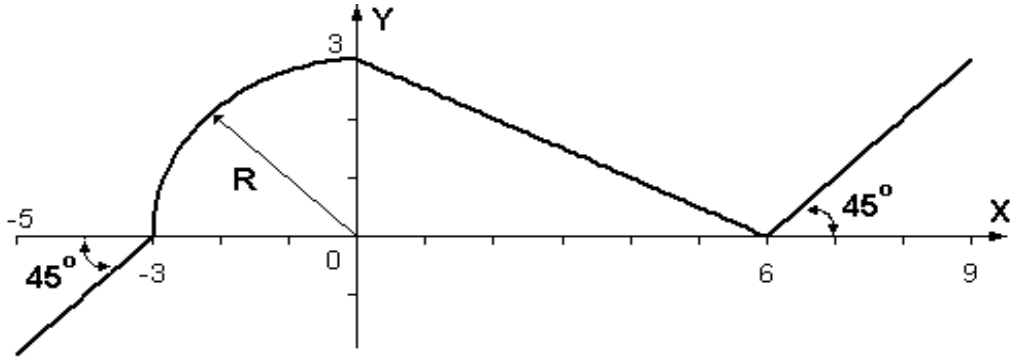
11)


Y
X
45°
R
1
0
-1
-3
-2
-1
1
2
3
4
5

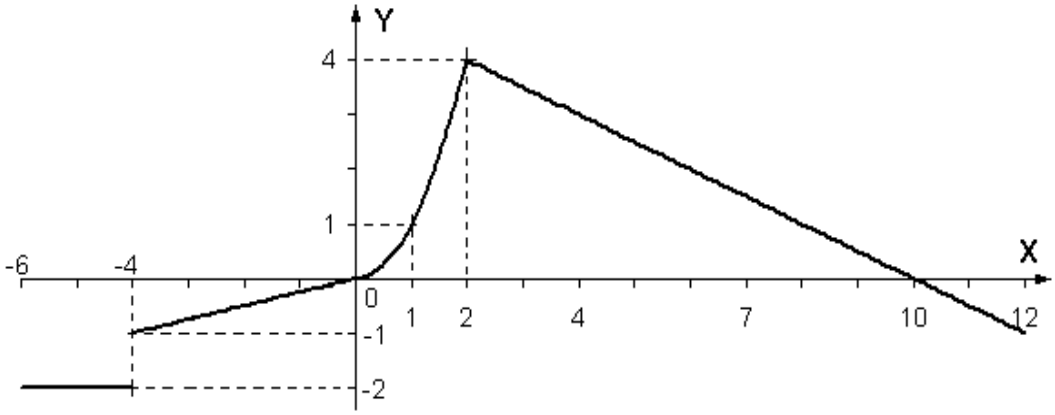
12)


Y
X
R
R
2
0
-1
-7
-6
-2
2
3

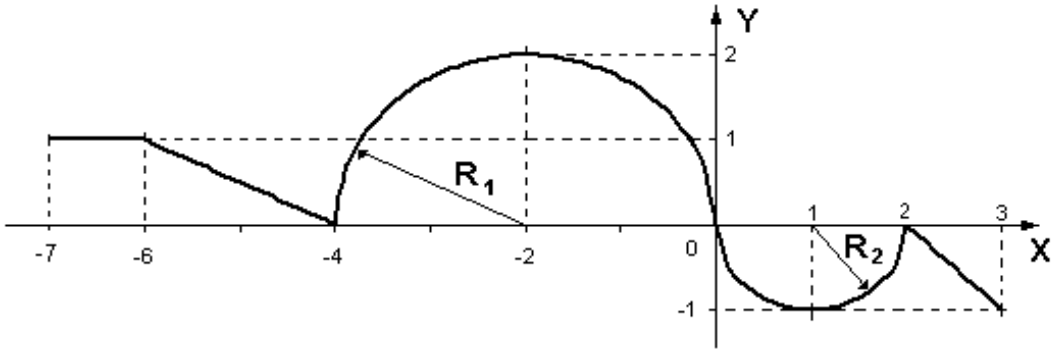
13)


Y
3
R
-5
45°
45°
X
-3
0
6
9

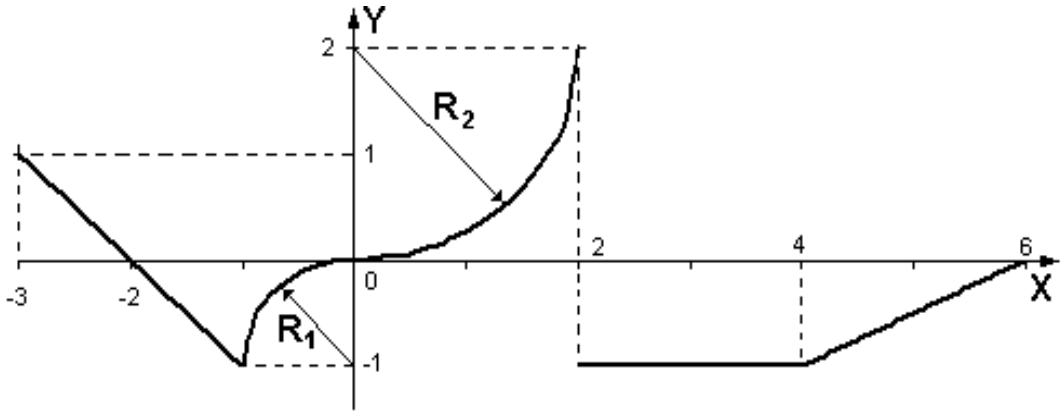
14)


Y
4
1
X
-6
-4
0
1
2
4
7
10
12
-1
-2

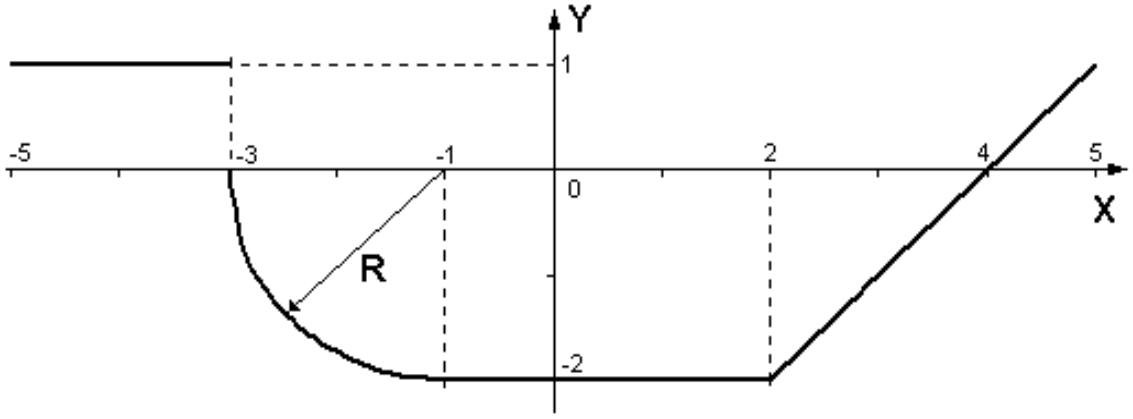
15)


Y
2
1
R1
-7
-6
-4
-2
0
1
2
3
R2
X
-1

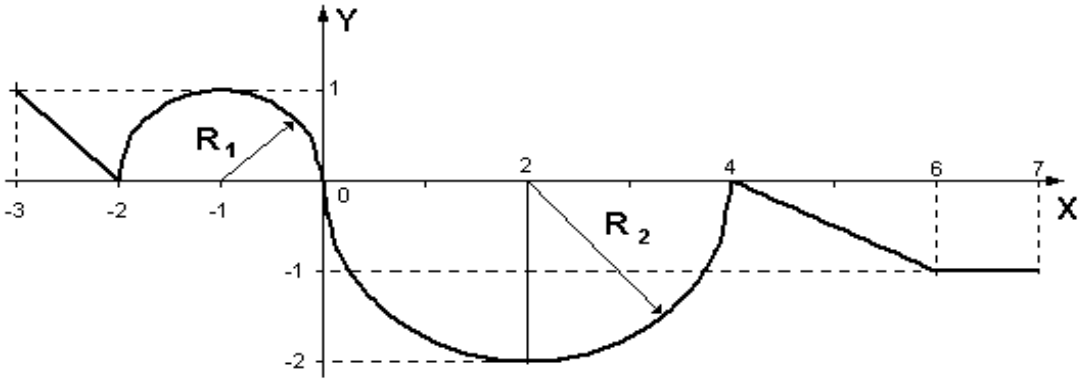
16)


Y
2
R2
1
2
4
6
-3
-2
0
X
R1
-1

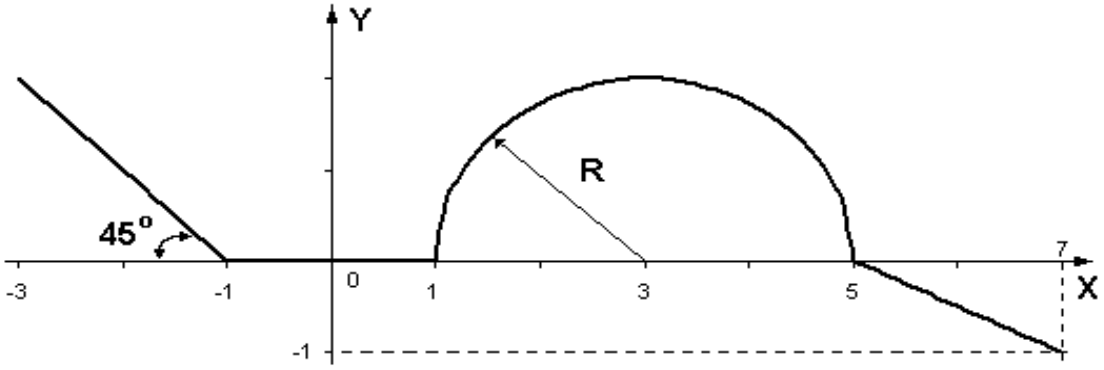
17)


Y
X
1
0
-2
-5
-3
-1
2
4
5
R

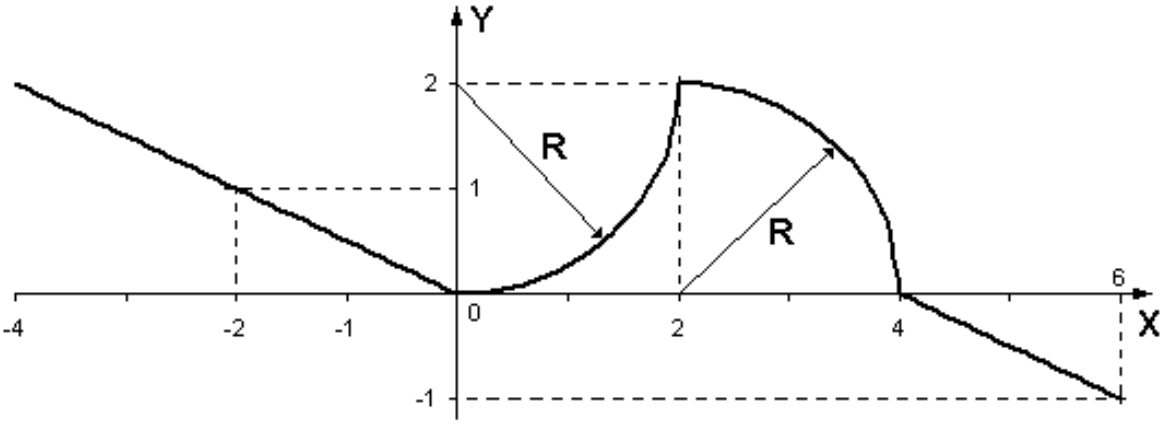
18)


Y
X
1
0
-1
-2
-3
-2
-1
2
4
6
7
R1
R2

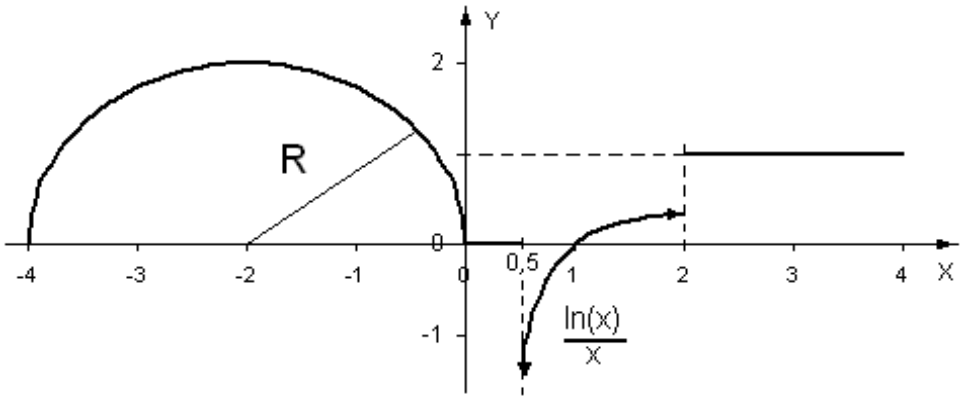
19)


Y
X
0
-1
-3
-1
1
3
5
7
R
45°

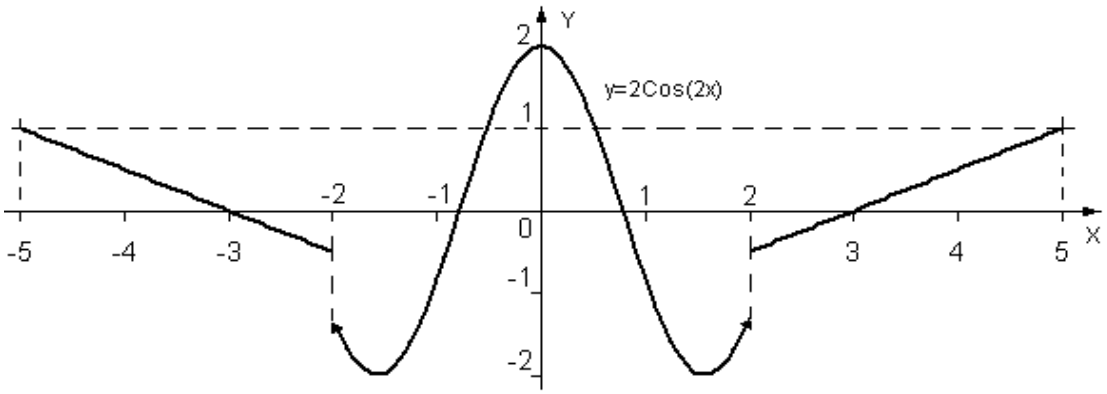
20)


Y
X
2
1
0
-1
-4
-2
-1
2
4
6
R
R

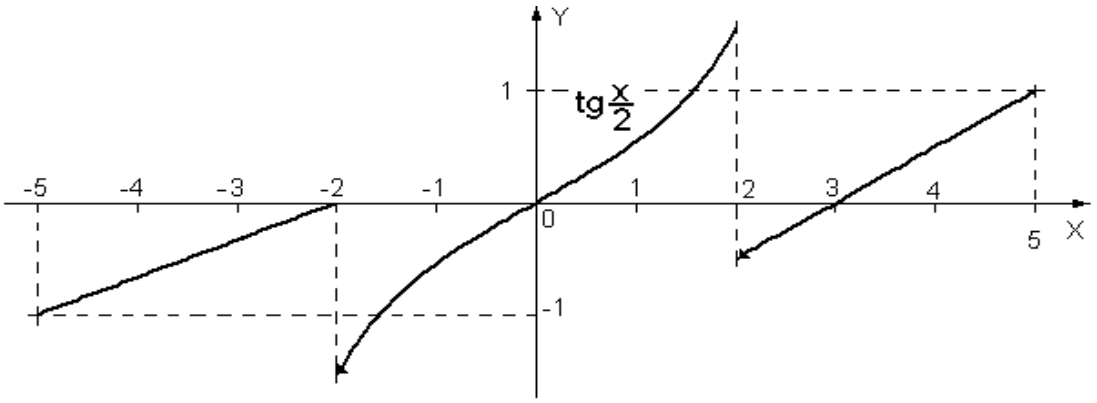
21)


Y
2
R
0
-4
-3
-2
-1
0
0,5
1
2
3
4
X
-1
ln(x)
x

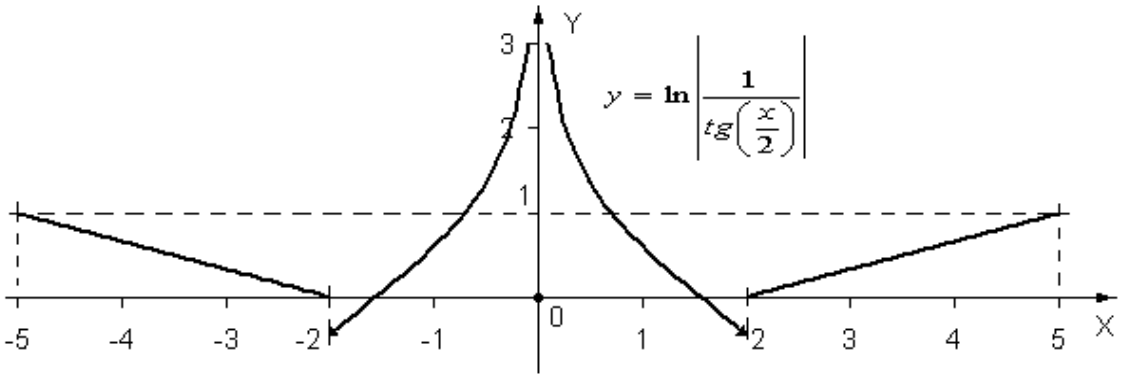
22)


Y
2
y=2Cos(2x)
1
-2
-1
1
2
-5
-4
-3
0
-1
-2
3
4
5
X

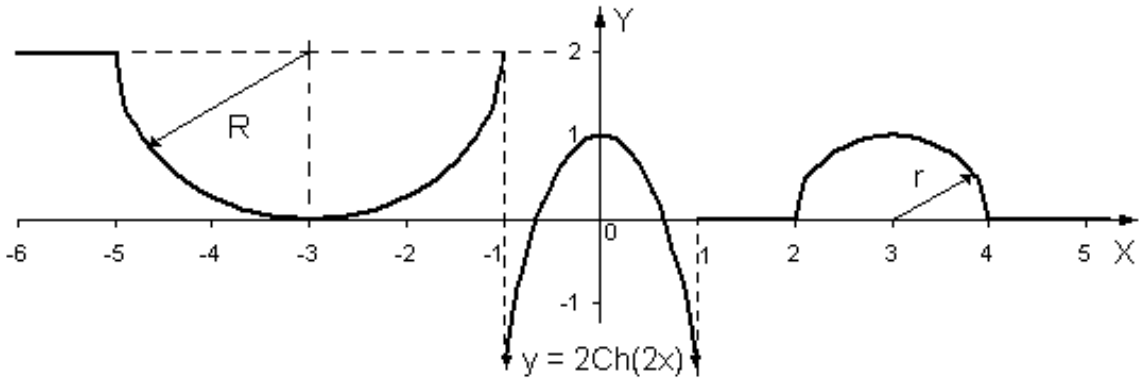
23)


Y
1
tg x/2
-5
-4
-3
-2
-1
1
2
3
4
0
5
X
-1

24)

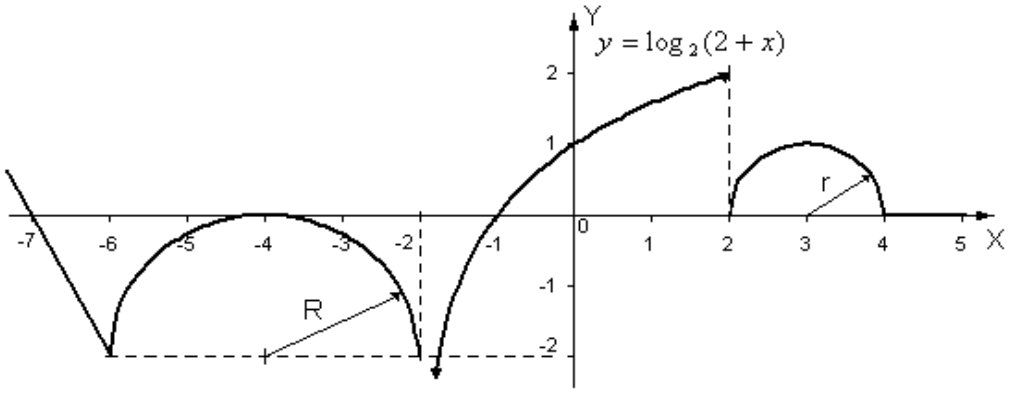

Y
3
y = ln |1 / tg(x/2)|
2
1
0
-5
-4
-3
-2
-1
1
2
3
4
5
X

25)

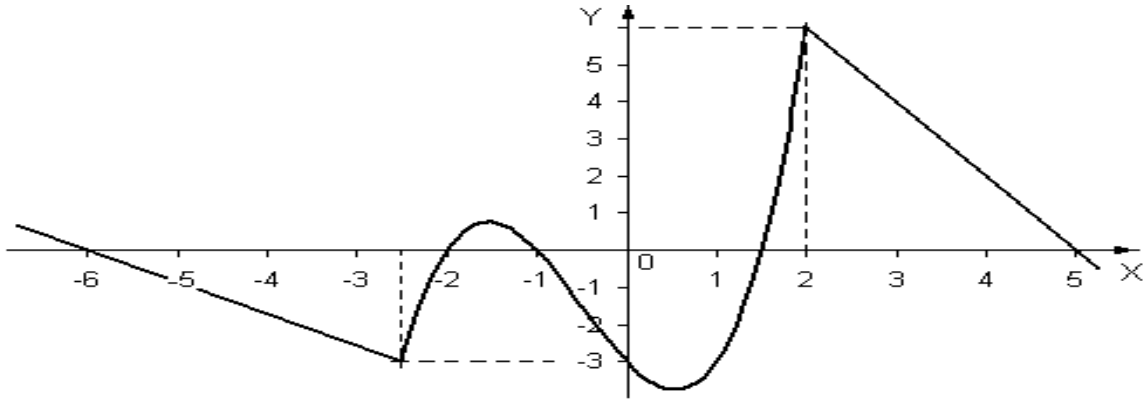


Гиперболический косинус может быть описан формулой: $Ch(x) = \frac{1}{2}\left(e^{x} + e^{-x}\right)$.

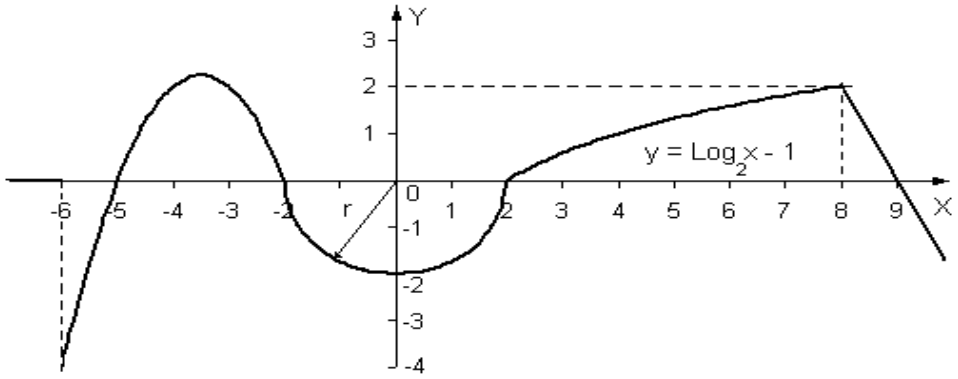
26)



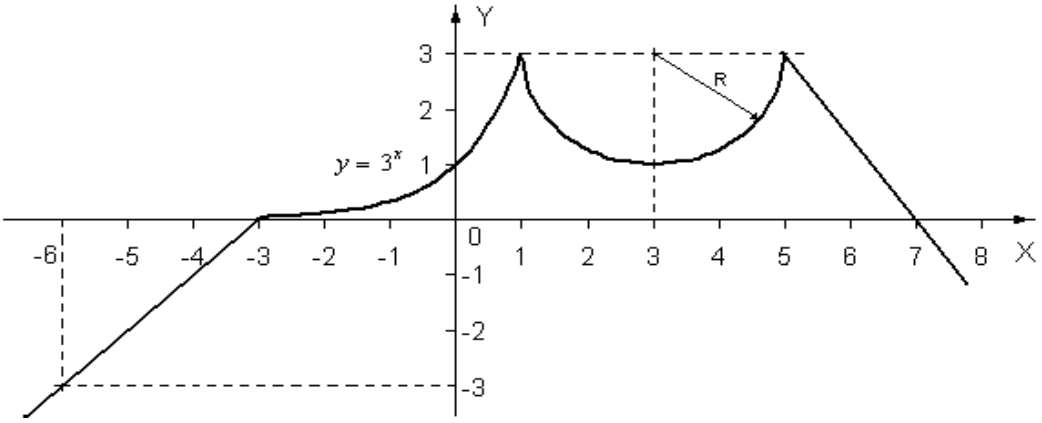
27)



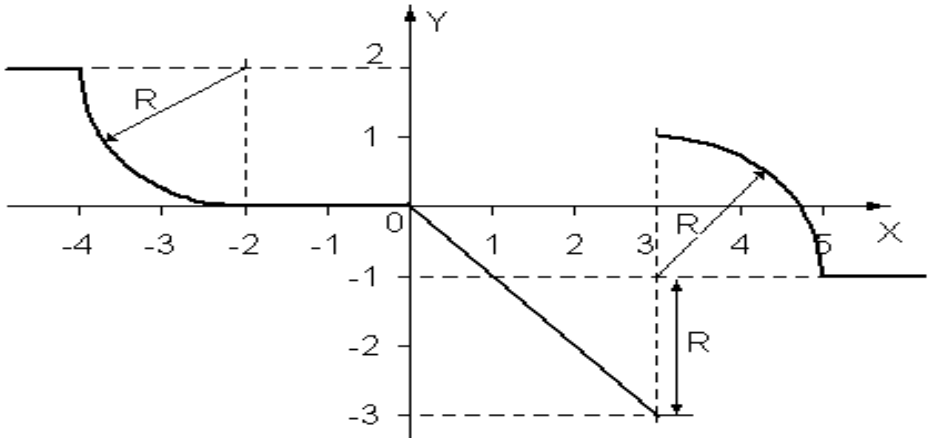
28)



29)



30)

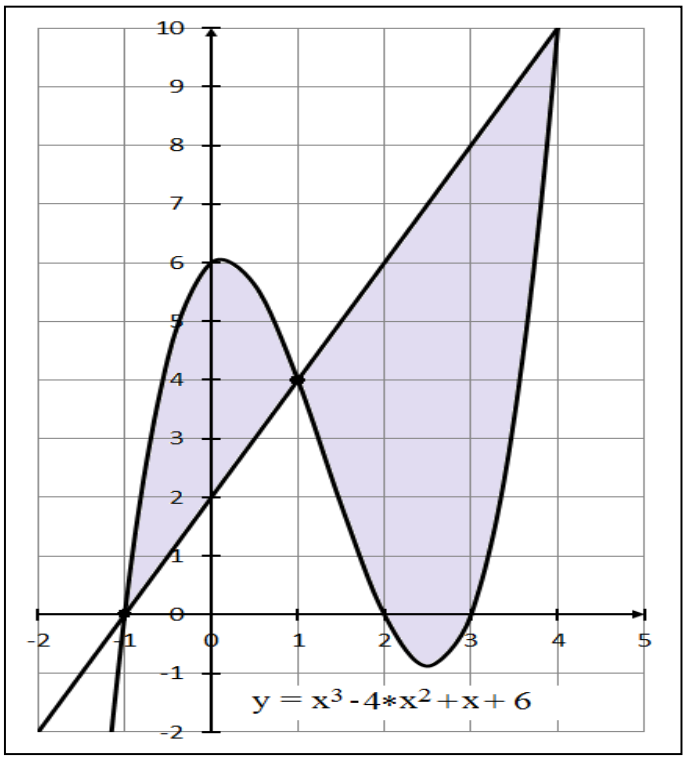


Дополнительное требование к заданию №30: Программа должна быть написана так, что бы решения получались при различных значениях `R`, вводимых с клавиатуры. Центр положения левой четверти окружности изменяется в соответствии с введённым радиусом, а правой – остаётся постоянным при `X = 3, Y = -1`.

Лабораторная работа №2: «Разветвляющиеся вычислительные процессы». Задание 2

Постановка задачи

Написать программу, которая определяет, попадает ли точка с заданными координатами в заштрихованную область. Точки на границе принадлежат области. Необходимые параметры получить из рисунка. Результат работы программы вывести в виде текстового сообщения: Попадает, Не попадает.



Теоретическое введение

Для решения задачи воспользуемся условным оператором:

```
if <Логическое выражение>:
    <Блок – выполняется, если условие истинно>
[elif <Логическое выражение>:
    <Блок – выполняется, если условие истинно>
]
[else:
     <Блок – выполняется, если все условия ложны>
]
```

<Блок> – это набор инструкций, которые выделяются одинаковым количеством пробелов (обычно четырьмя).

Необходимо правильно составить логическое выражение, параметрами которого будут значение координат точки (x,y) и уравнения линий.

Обмен с консолью выполняется стандартными функциями ввода/вывода: `input()` и `print()`.

Для решения задачи требуется знать уравнение прямой (уравнение кривой приведено на рисунке). Из рисунка получаем координаты двух точек, через которые проходит прямая линия (-1, 0) и (1,4). Подставив значения выбранных точек в уравнение общего вида `y = kx + b`, получим систему уравнений. Решив эту систему, найдём значения для параметров k и b. В итоге уравнение прямой примет вид: `y = 2x + 2`.

Точка с координатами (`x, y`), введённая пользователем, будет попадать в заштрихованную область на интервале [-1, 1] в том случае, если `x` будет не меньше -1 и не больше 1. Кроме этого, `y` должен быть не меньше значения полученного для прямой в точке `x` и не больше значения, вычисленного для кубической параболы в этой точке.

Описание алгоритма

1. Ввести координаты точки (x, y) и привести значения к типу `float`.
2. Выполнить проверку на попадание точки в заданную область.
3. Вывести результат в виде: "Точка `x, y` попадает в область." и "Точка `x, y` не попадает в область."

Описание входных и выходных данных

Входные данные - координаты точки, введённые пользователем. Тип данных и точность представления в задаче не заданы. Установим вещественный тип (`float`).

Выходные данные - сообщения, в текстовом виде, о попадании или непопадании точки в заданную область.

Тестовые примеры

X	Y	Результат
-1	0	Попадает
-0.5	-1	Не попадает
0	3	Попадает
1	4	Попадает
1	5	Не попадает
1.5	1	Не попадает
2	3	Попадает
2.5	-1	Не попадает
2.5	-0.3	Попадает
3.5	10	Не попадает

Листинг программы

```
# -*- coding: cp1251 -*-
from math import *
flag = 0
print('Введите координаты X и Y для точки:')
x = float(input('X='))
y = float(input('Y='))
```

```
if (x < -1) or (x > 4):
    flag = 0        #False
if ((x>=-1) and (x<1) and (y>=2*x+2)
    and (y<=x**3-4*x**2+x+6) or
    (x>=1) and (x<=4) and (y>=x**3-4*x**2+x+6)
    and (y<=2*x+2)):
    flag = 1
else:
    flag = 0
print("Точка X={0: 6.2f} Y={1: 6.2f}"
      .format(x, y), end=" ")
if flag:
    print("попадает", end=" ")
else:
    print("не попадает", end=" ")
print("в область.")
```

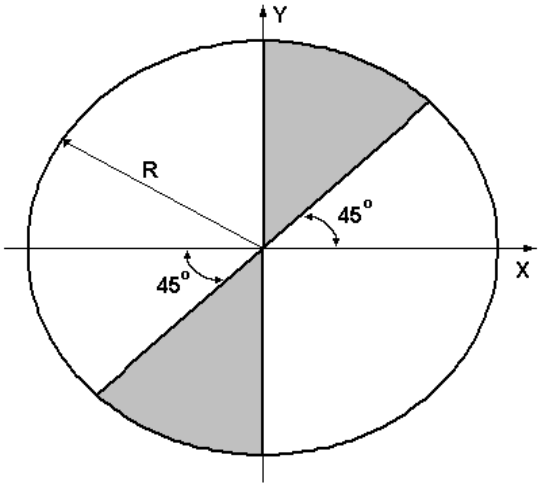
Замечание к тексту программы: Для переноса длинных строк в Python использованы круглые скобки.

Задание к лабораторной работе №2 «Разветвляющиеся вычислительные процессы».

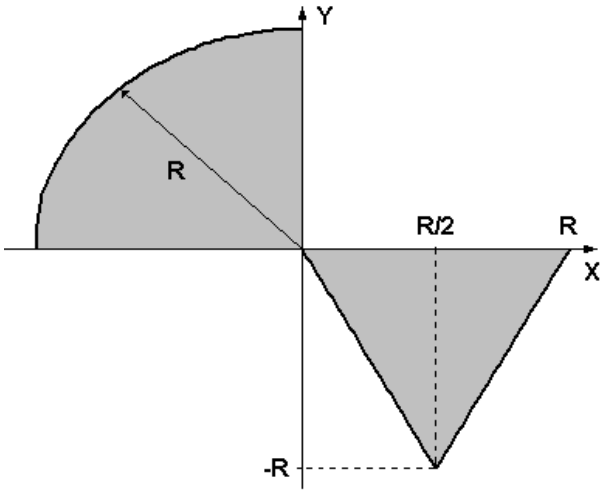
Задание 2

Написать программу, которая определяет, попадает ли точка с заданными координатами *X, Y* в область, закрашенную на рисунке серым цветом. Результат работы программы вывести в виде текстового сообщения. Параметр *R* вводится с клавиатуры.

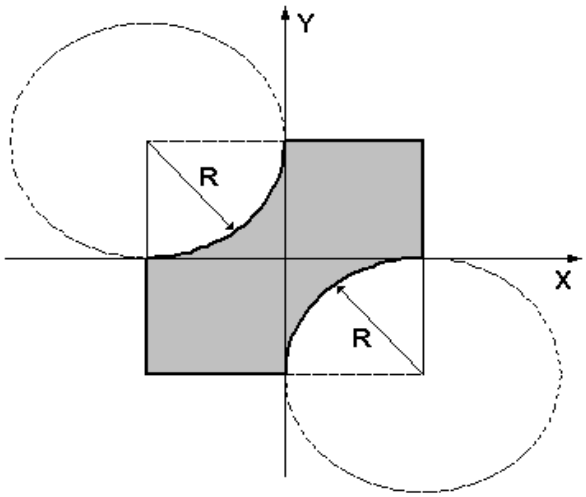
1)



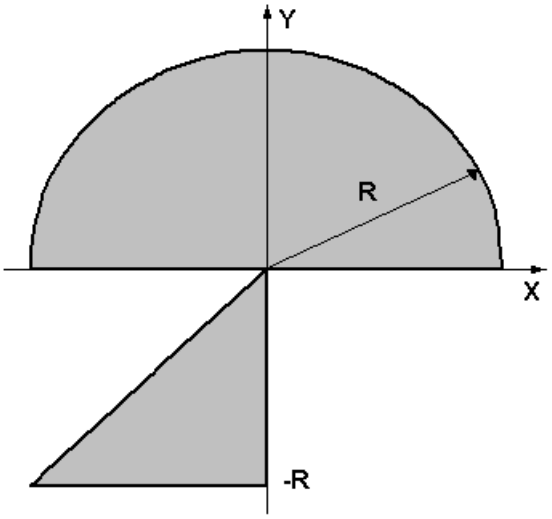
2)



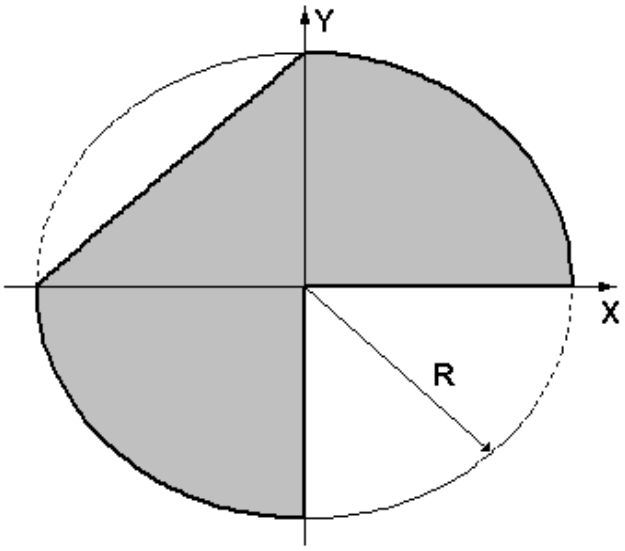
3)



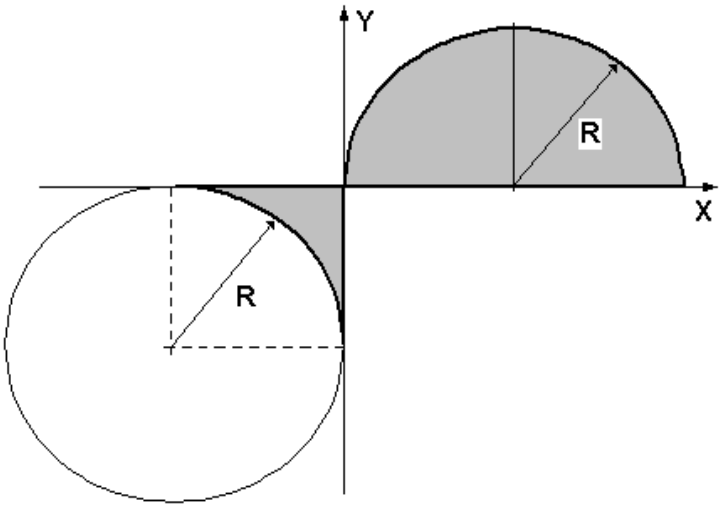
4)



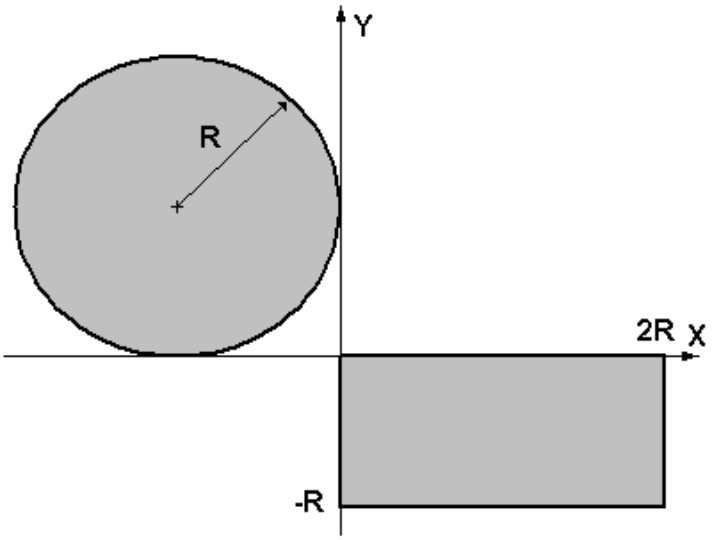
5)



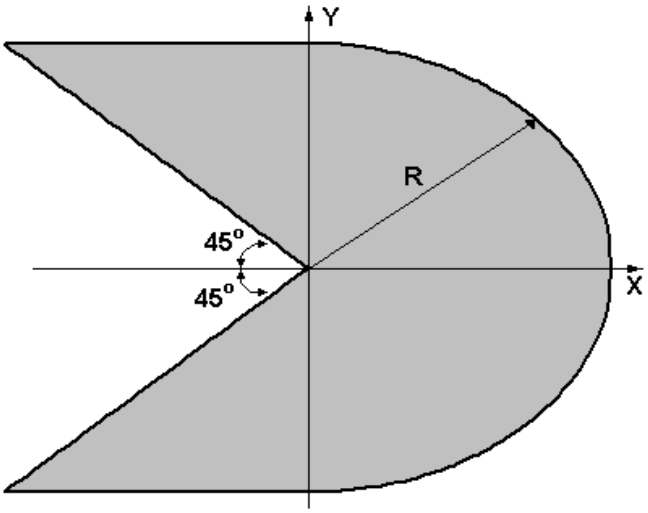
6)



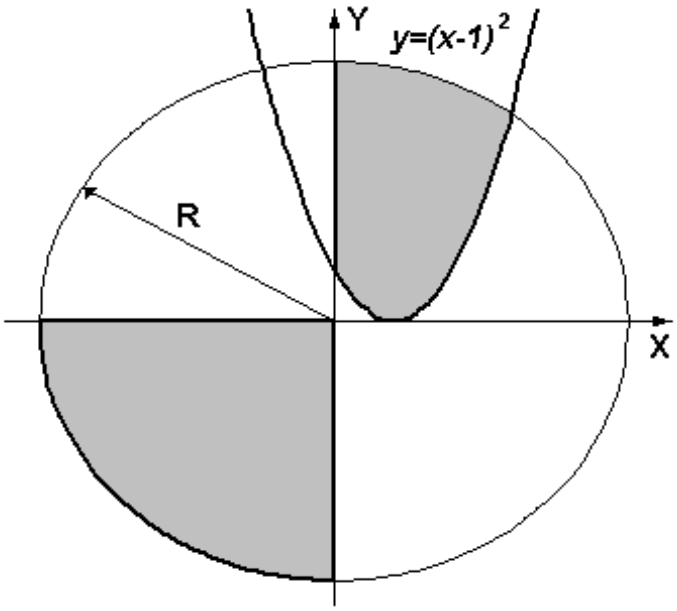
7)



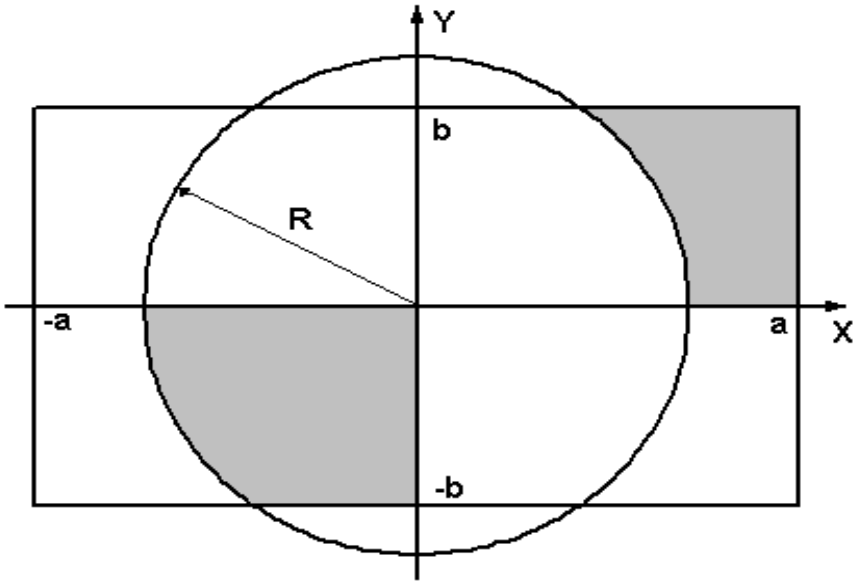
8)



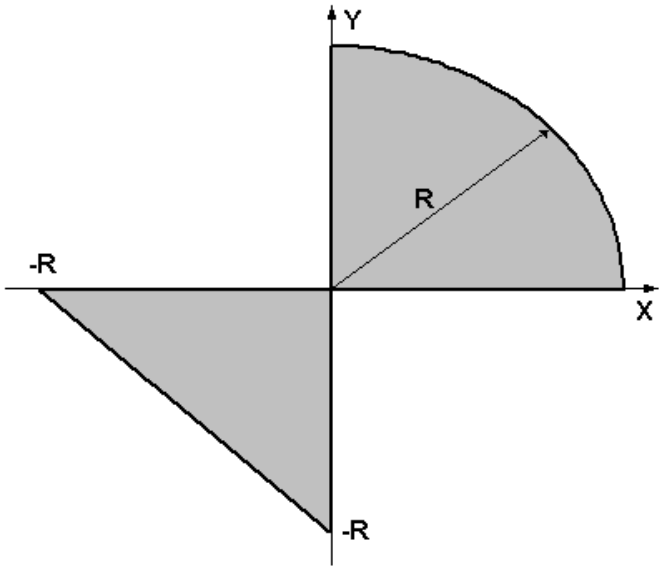
9)



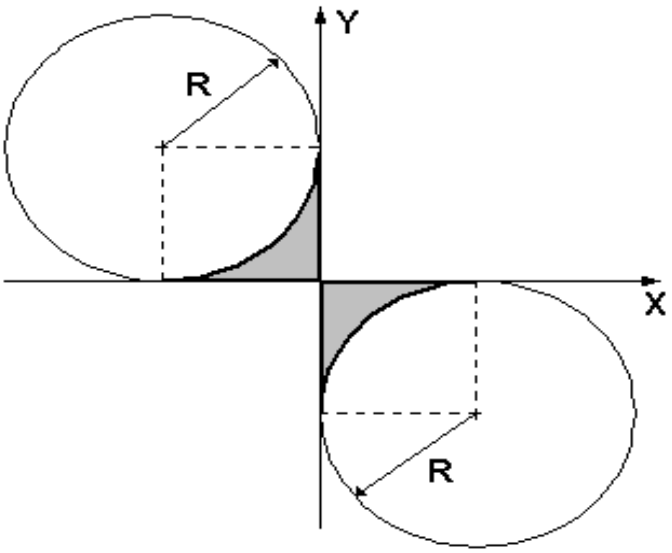
10)



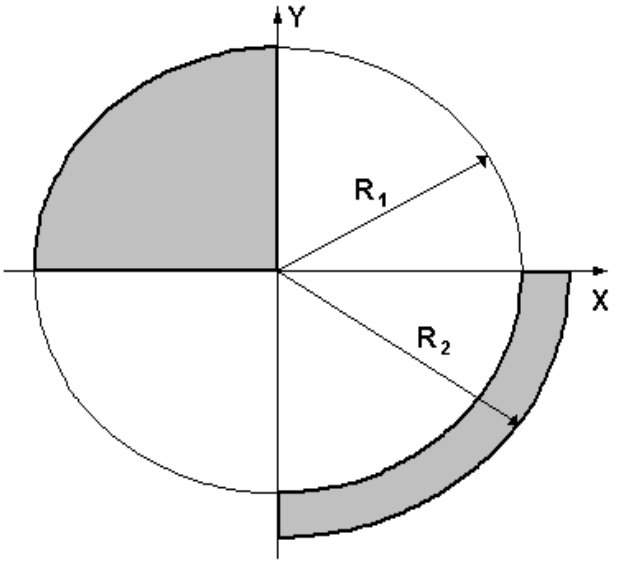
11)



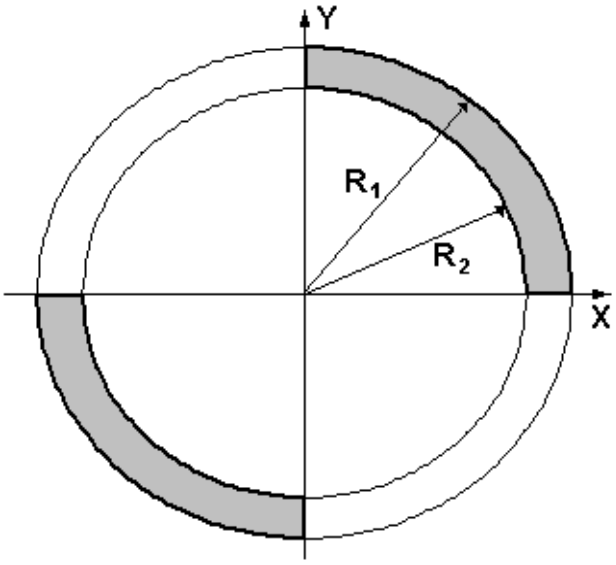
12)



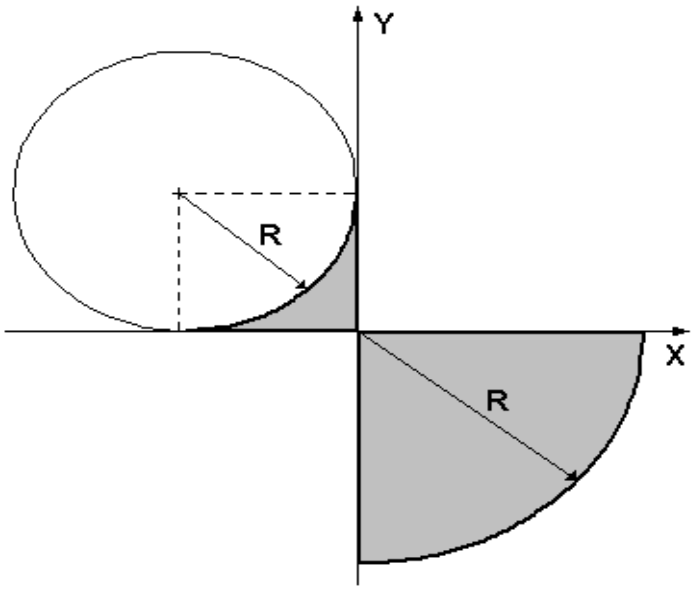
13)



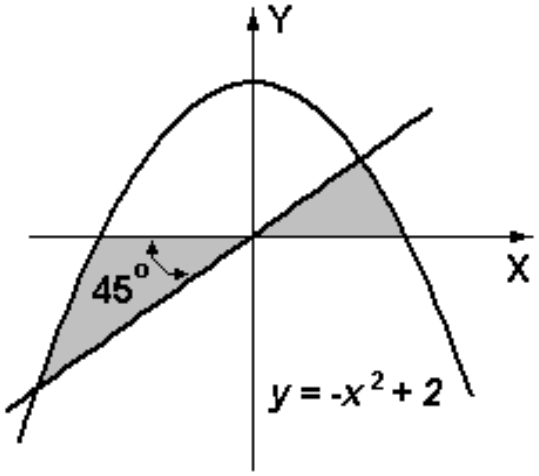
14)



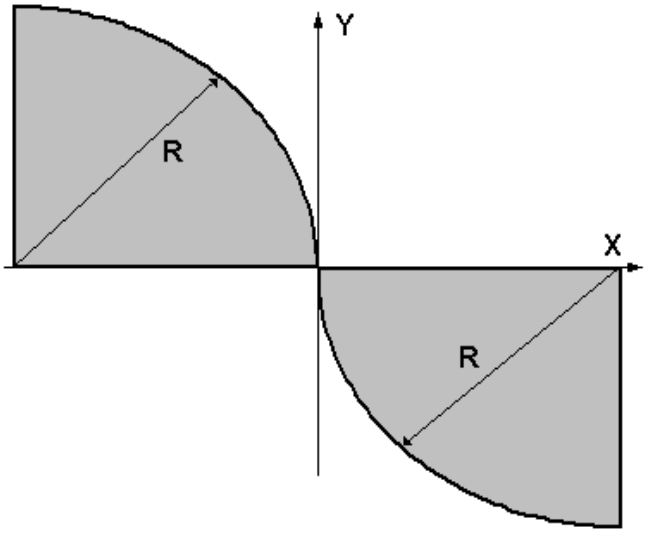
15)



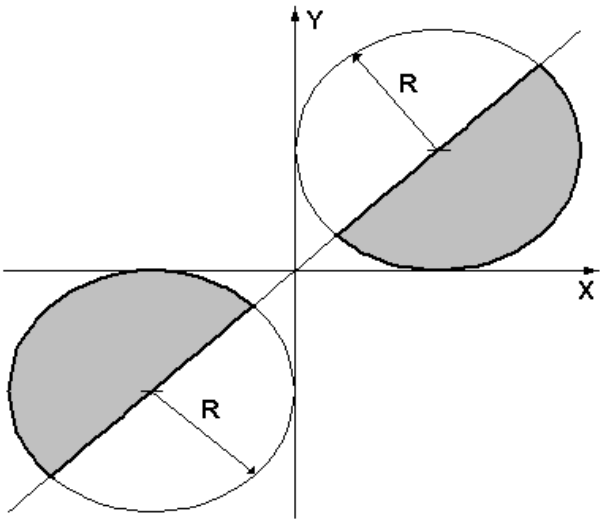
16)



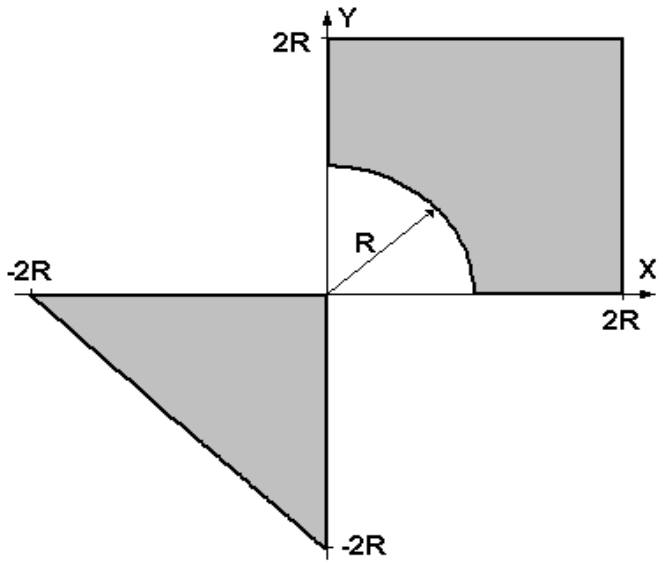
17)



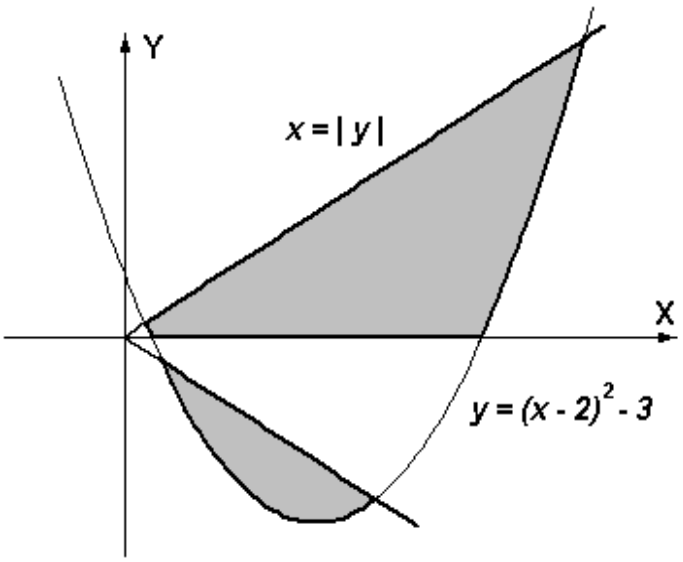
18)



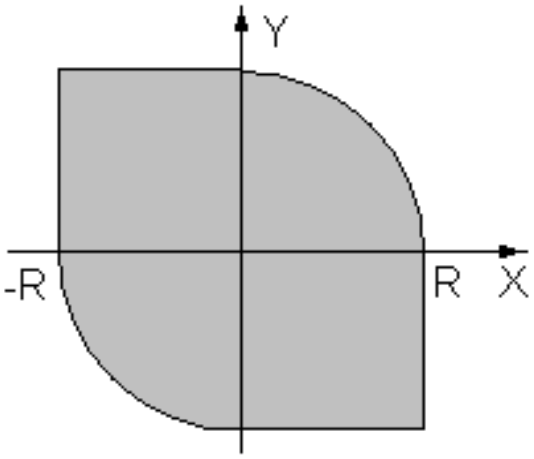
19)



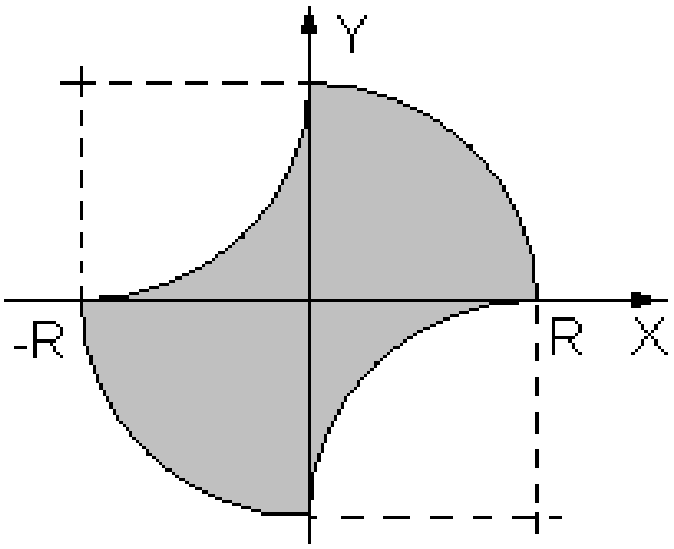
20)



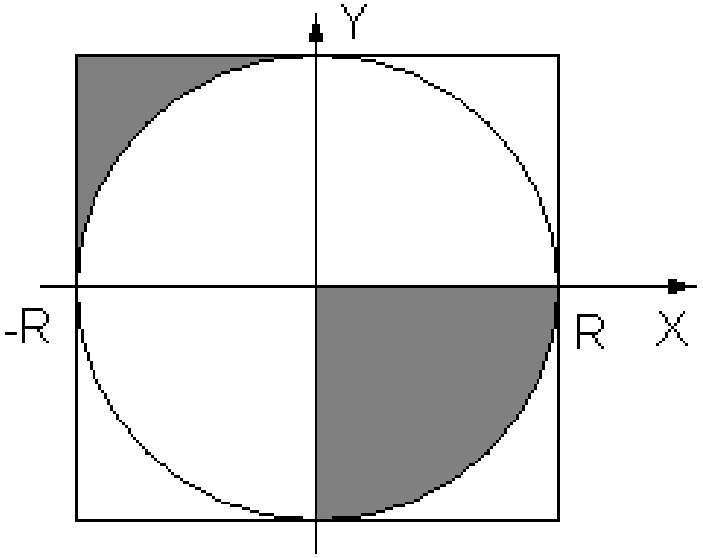
21)



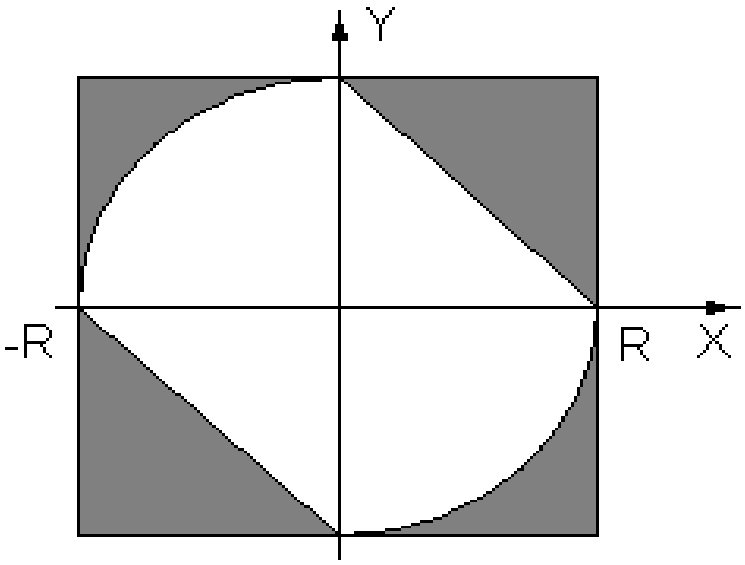
22)



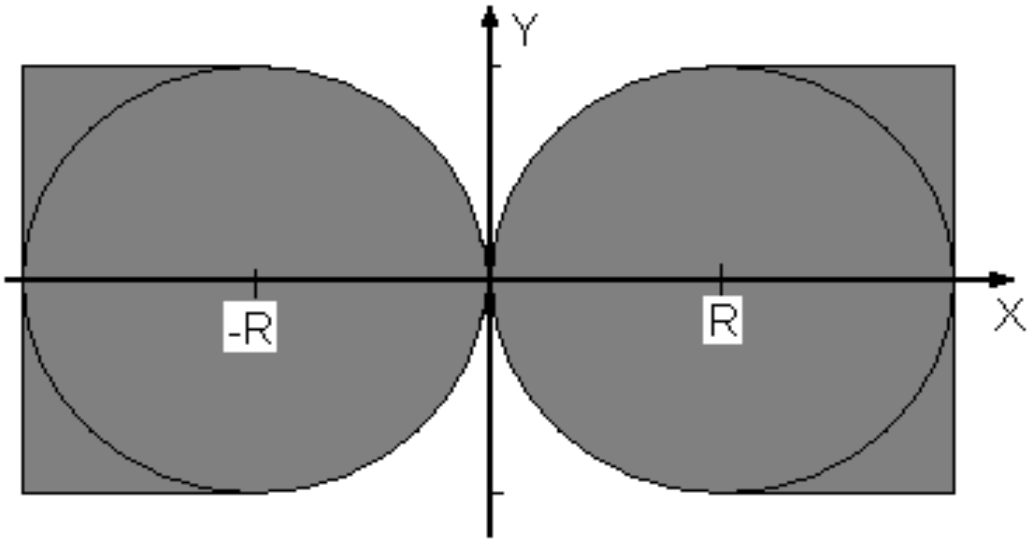
23)



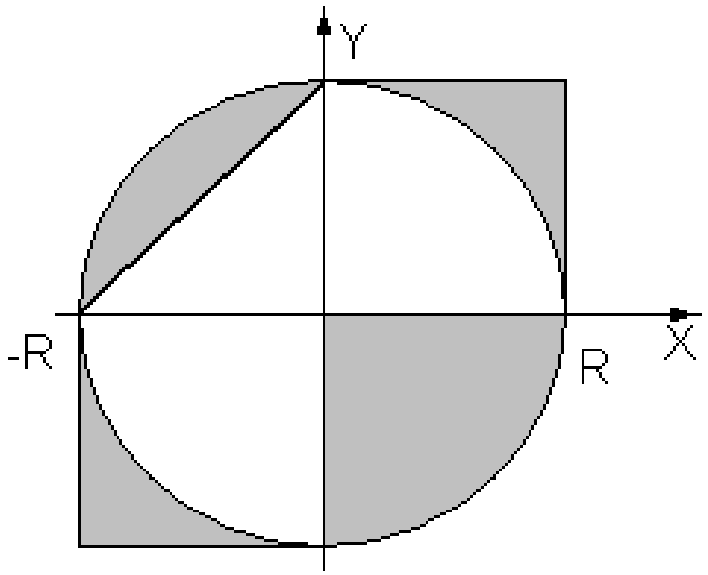
24)



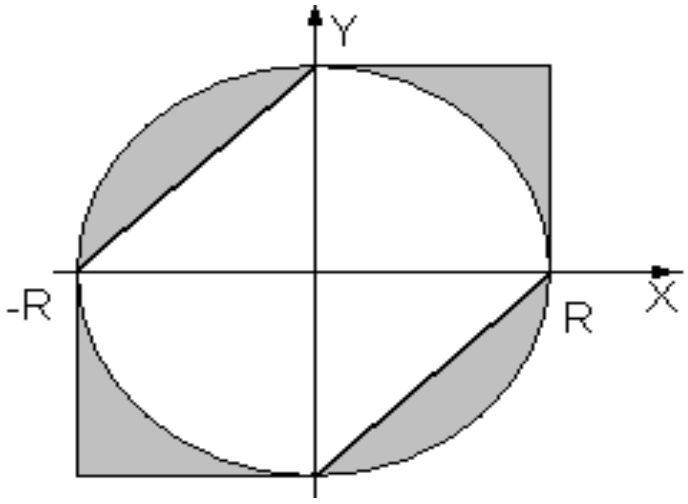
25)



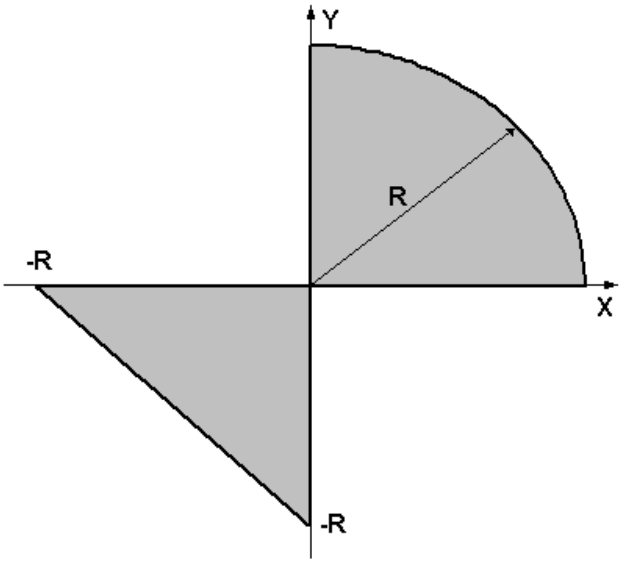
26)



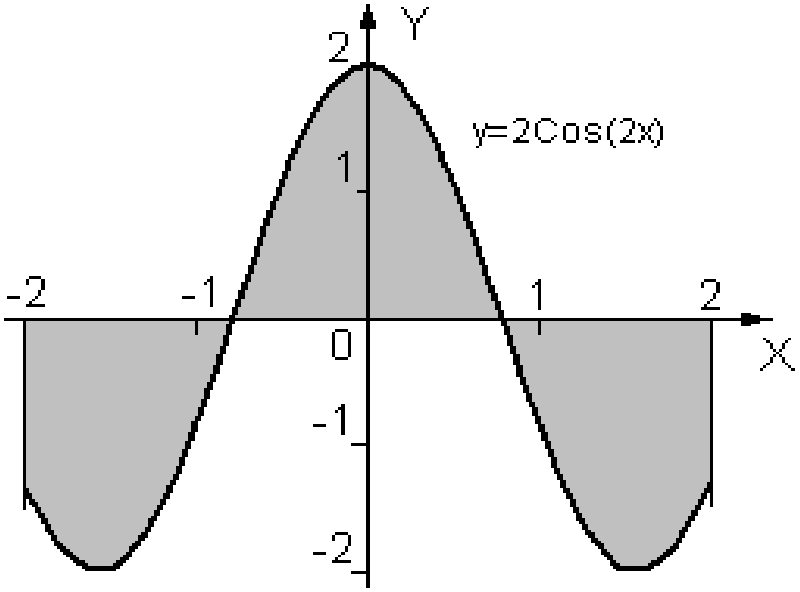
27)



28)



29)



30)