

Лабораторная работа №2: "Разветвляющиеся вычислительные процессы". Задание 1. Пример оформления.

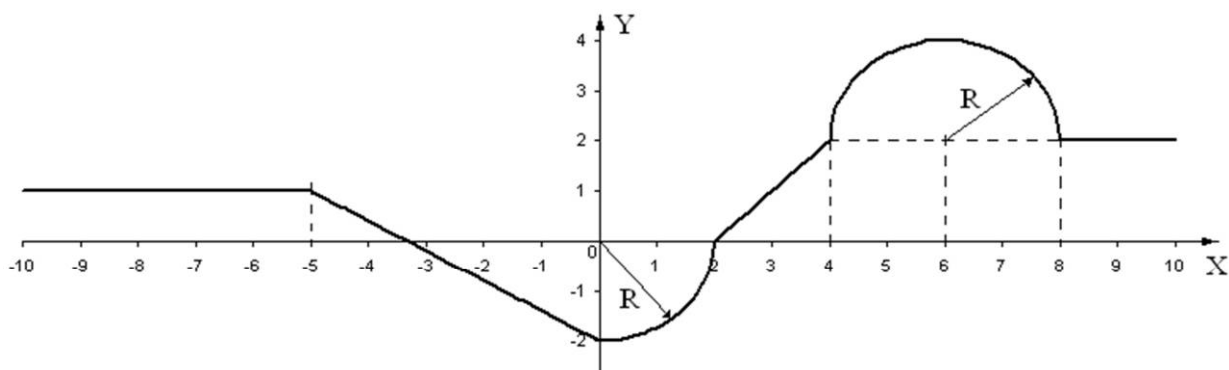
Цель работы

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

Следующий материал представляет пример оформления лабораторной работы.

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 инструкций, которые выделяются одинаковым количеством пробелов (обычно четырем).

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

Вывод данных выполняется инструкцией `print()`, в которой можно использовать форматирование выводимых данных.

Функция представлена фрагментами прямых линий, описываемых уравнением $y = kx + b$ и дугами кругов, обобщенное уравнение которых: $(x - a)^2 + (y - b)^2 = R^2$. Известные параметры, угол наклона и смещение прямой, а так же координаты центра дуг, определим, используя данные из графика. В итоге функция примет вид:

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

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

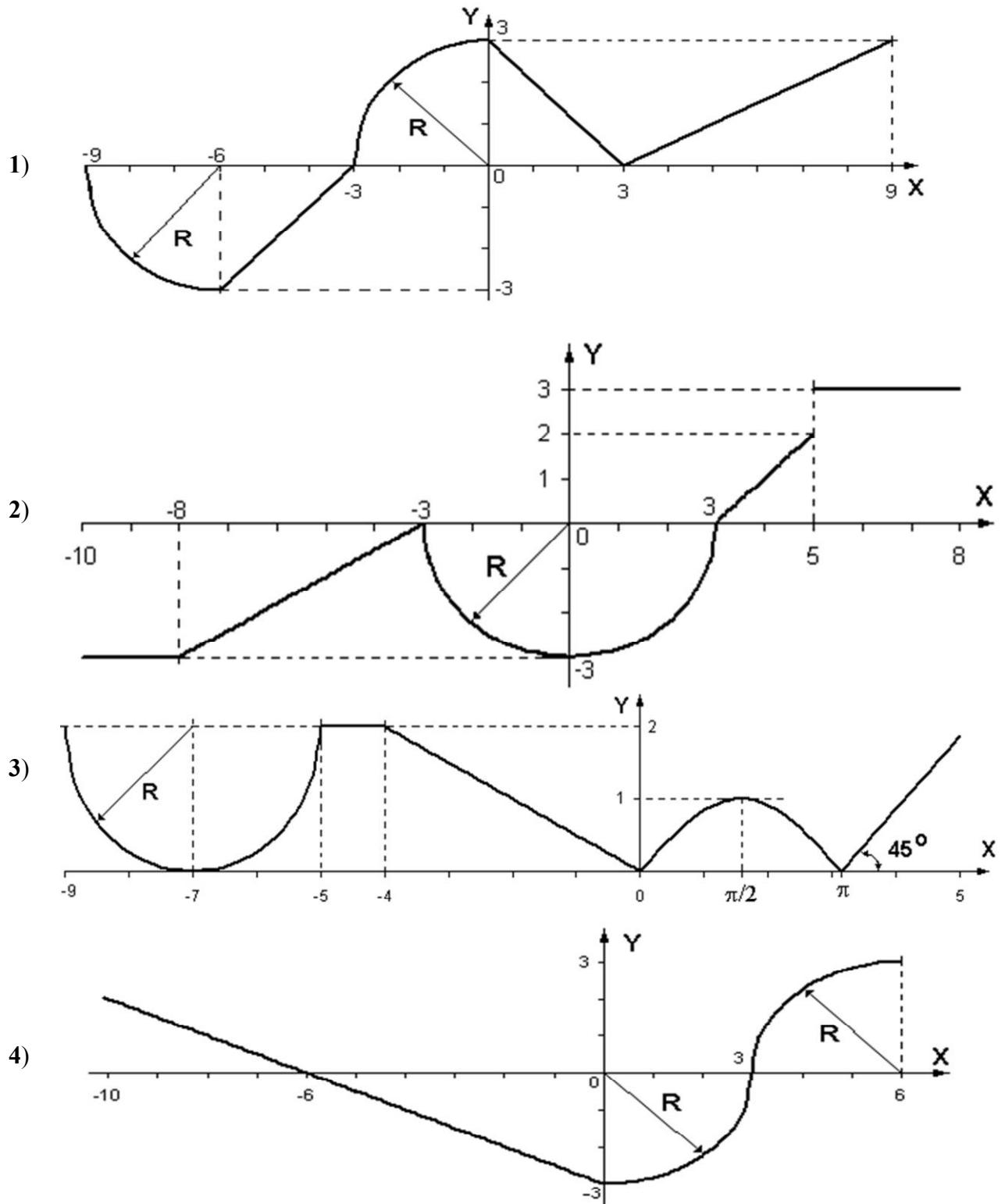
```
Введите значение аргумента: -6
X= -6.00      Y= 1
Введите значение аргумента: -3.33
X= -3.33      Y= -0.00
Введите значение аргумента: 6
X= 6.00       Y= 4.00
```

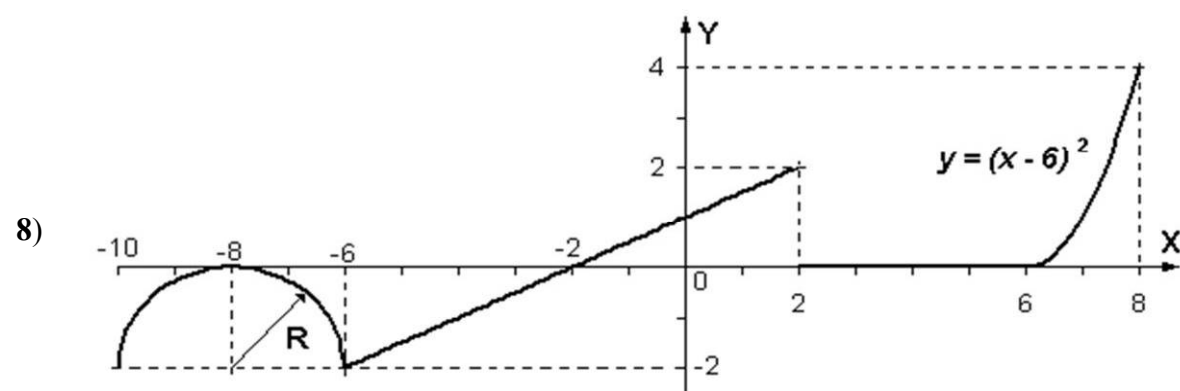
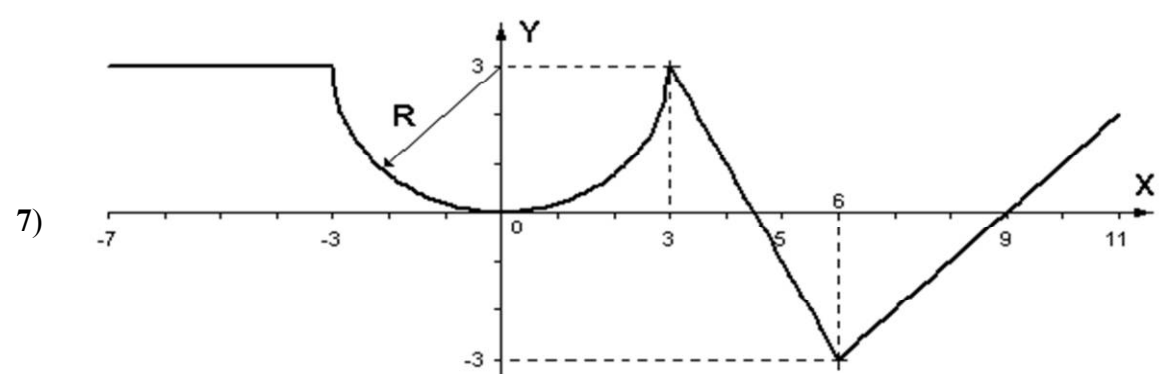
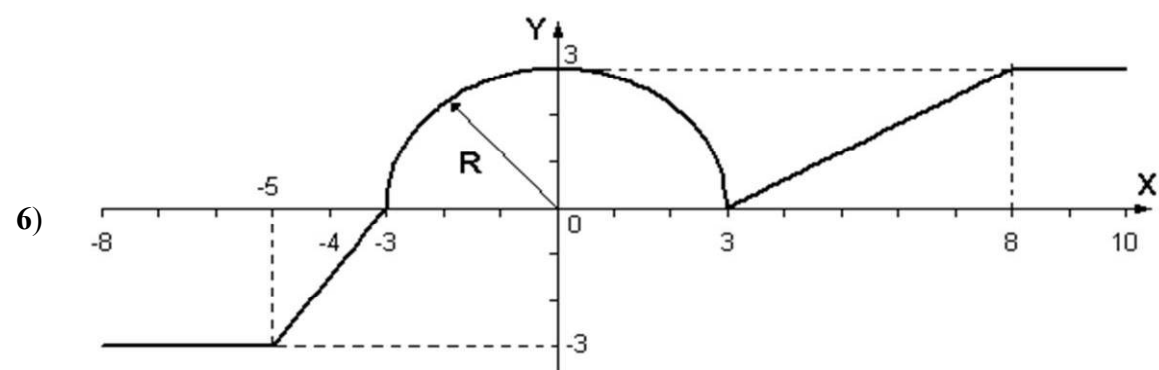
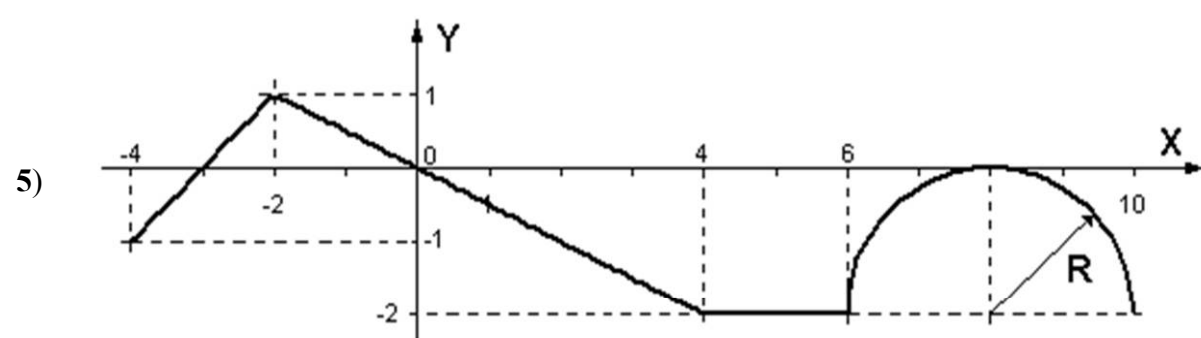
Список используемой литературы

1. Н.А. Прохоренок, В.А. Дронов, Python 3 и PyQt 5. Разработка приложений: СПб.: БХВ-Петербург, 2017
2. В.П. Рядченко, Методическое пособие по выполнению лабораторных работ.

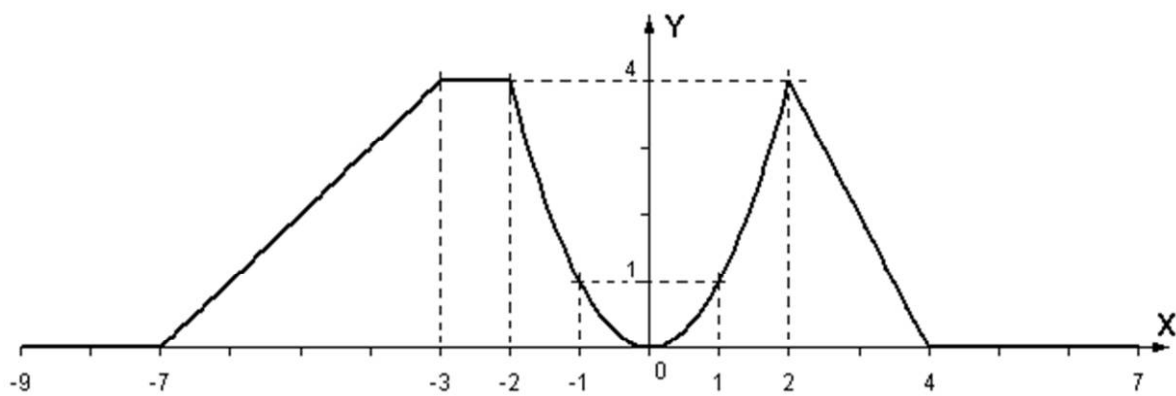
Задание к лабораторной работе №2 "Разветвляющиеся вычислительные процессы". Задание 1

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.

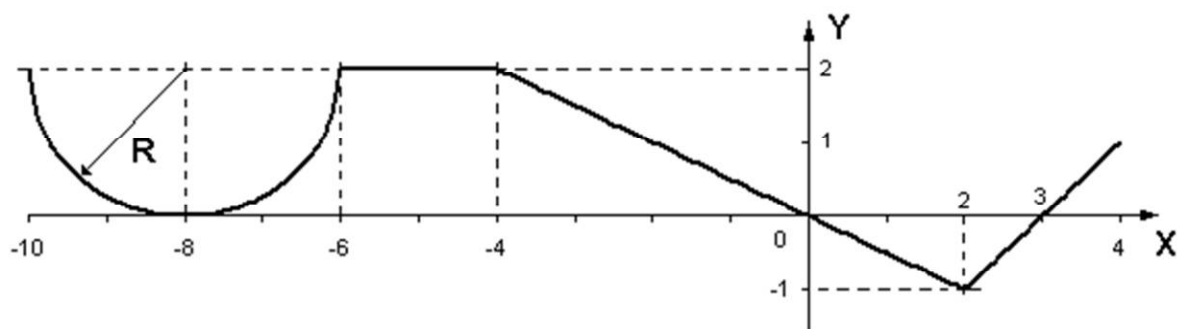




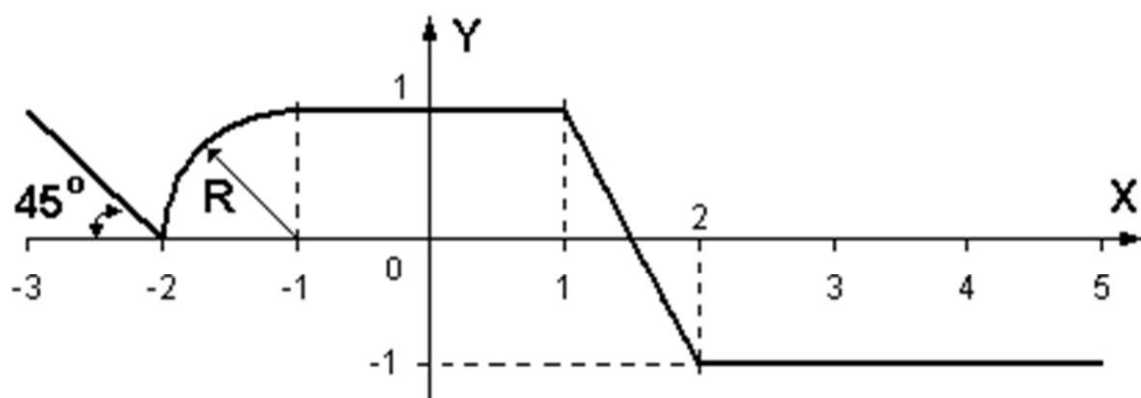
9)



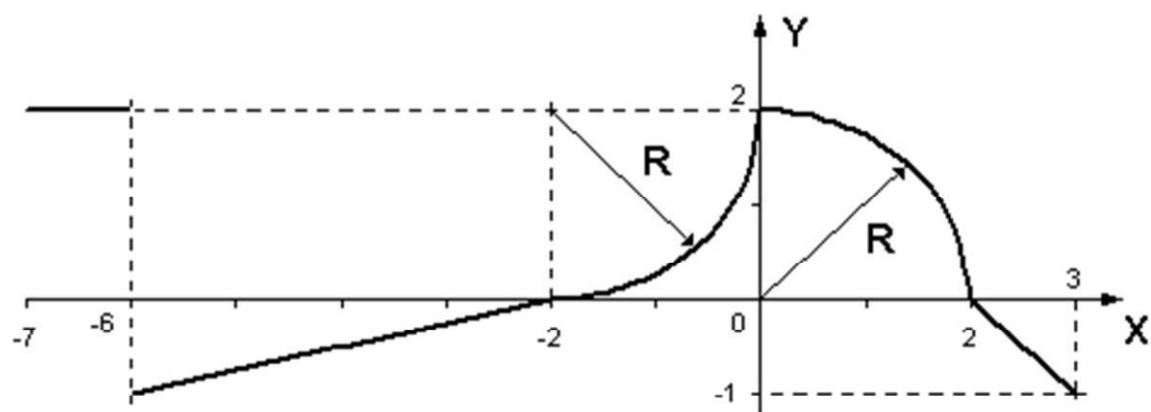
10)



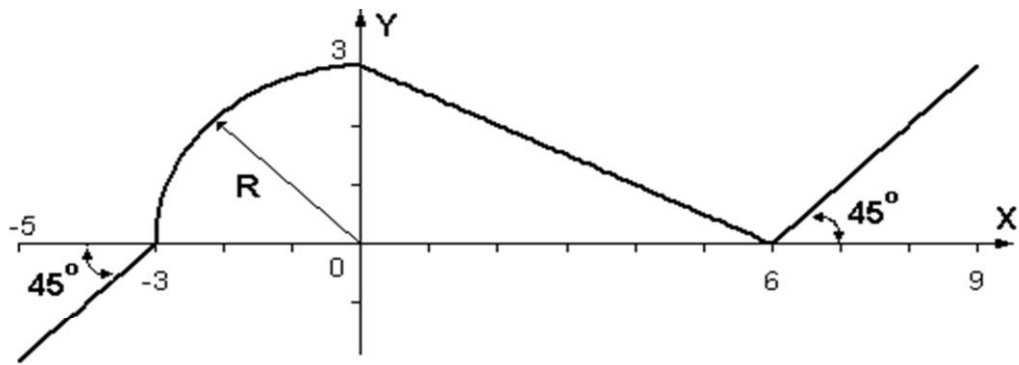
11)



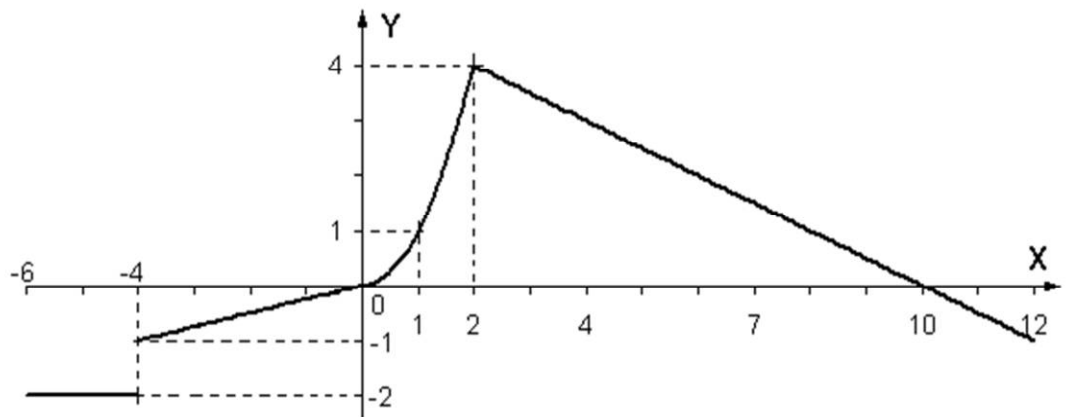
12)



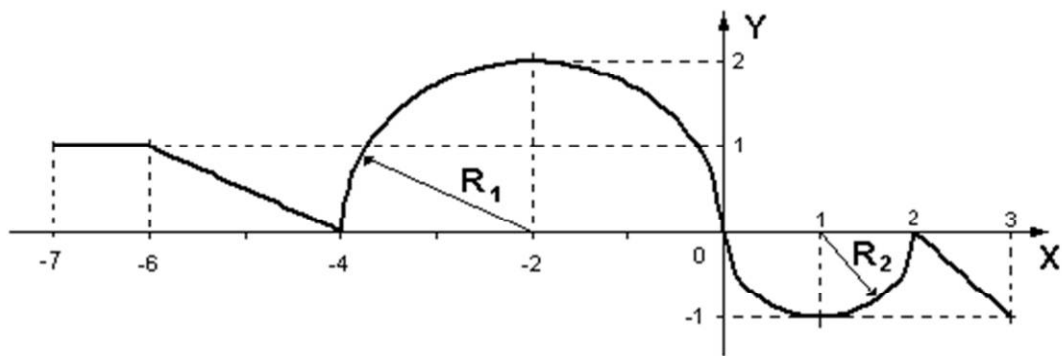
13)



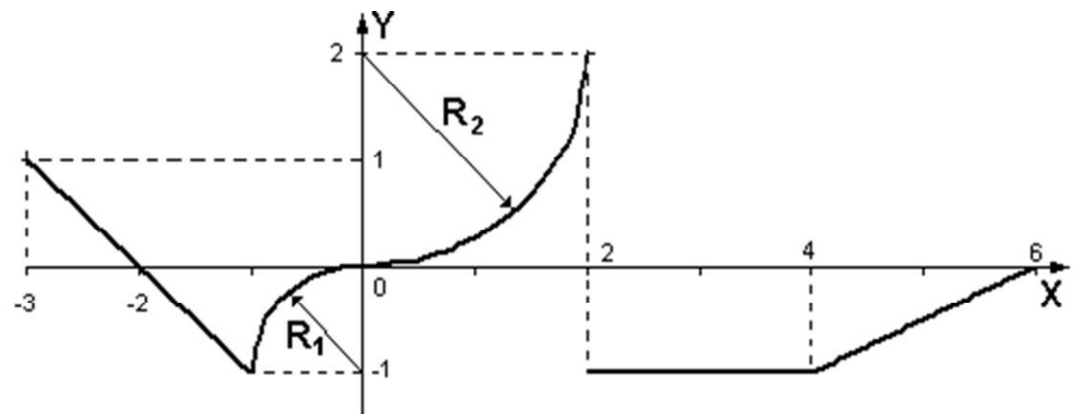
14)



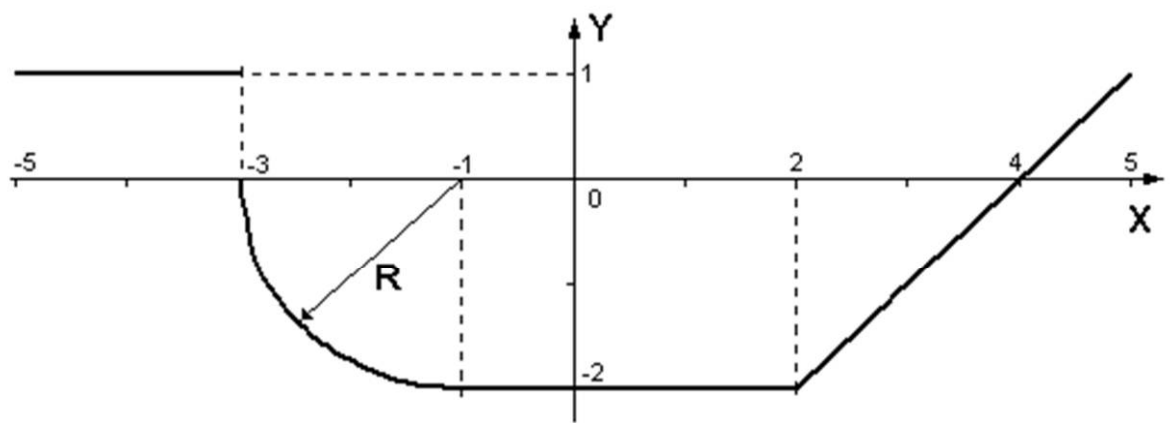
15)



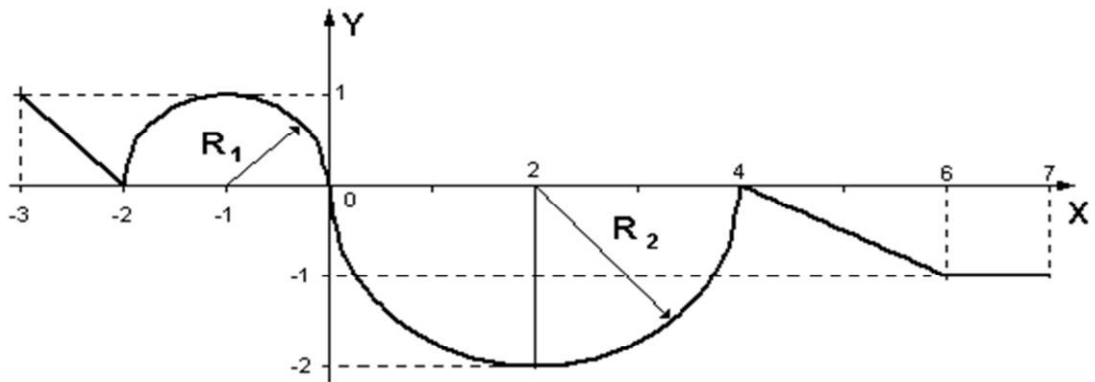
16)



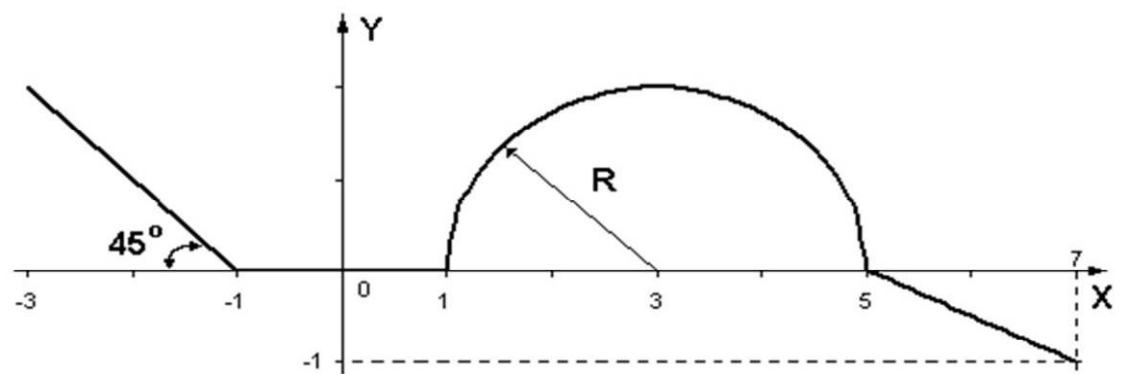
17)



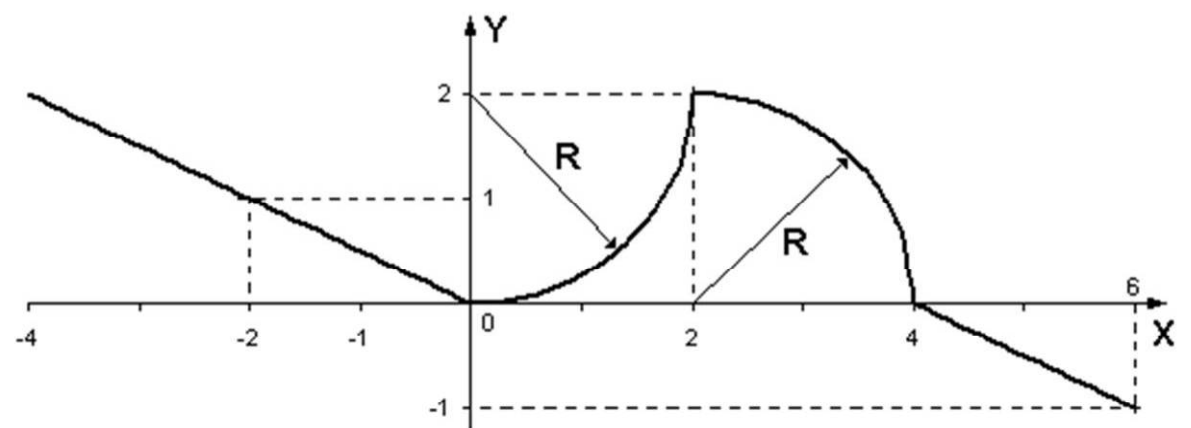
18)



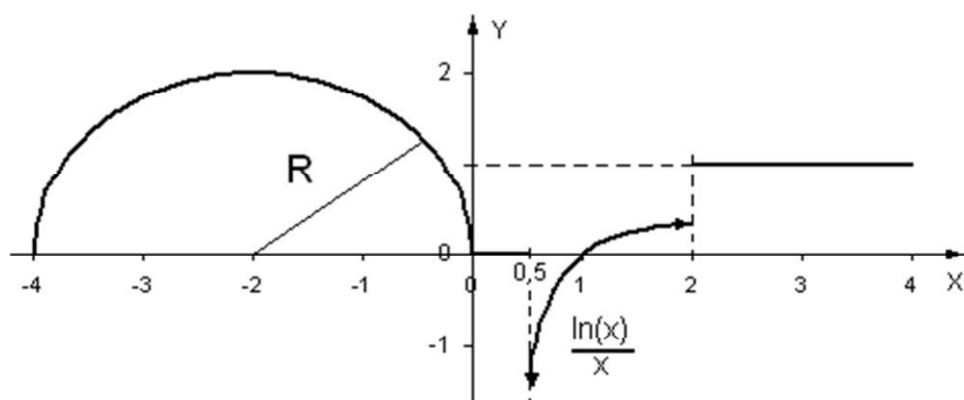
19)



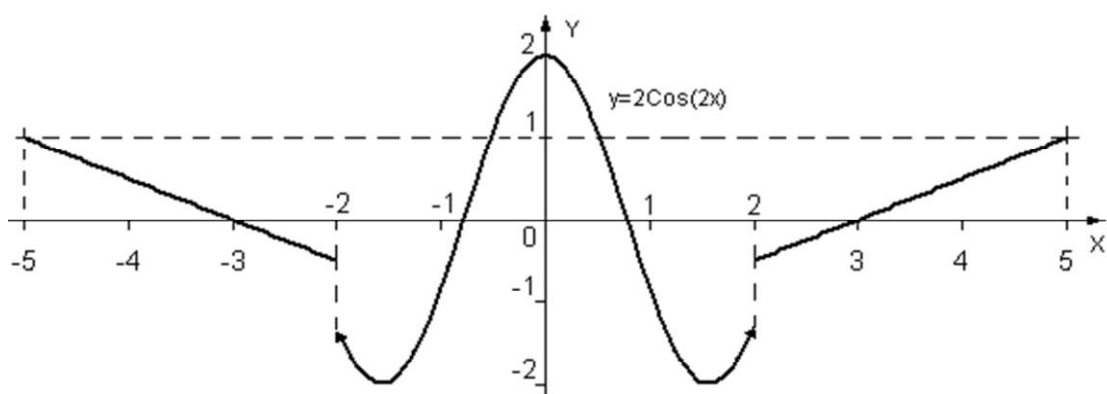
20)



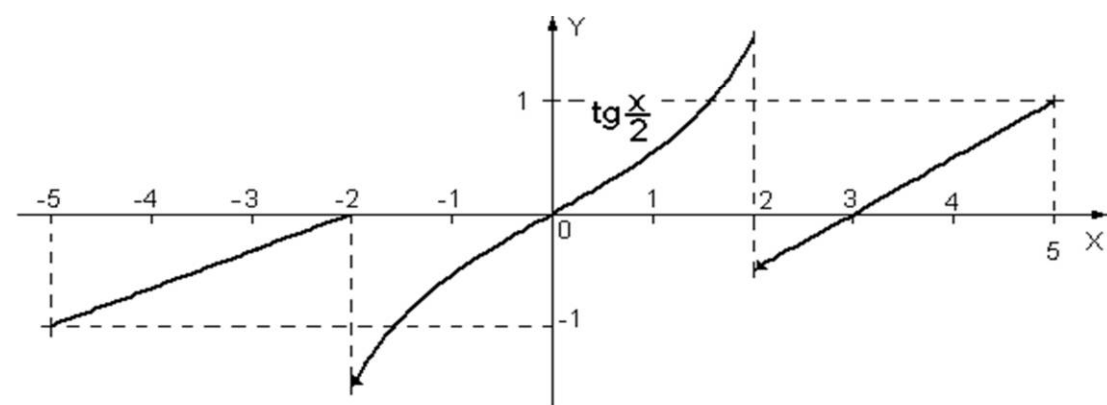
21)



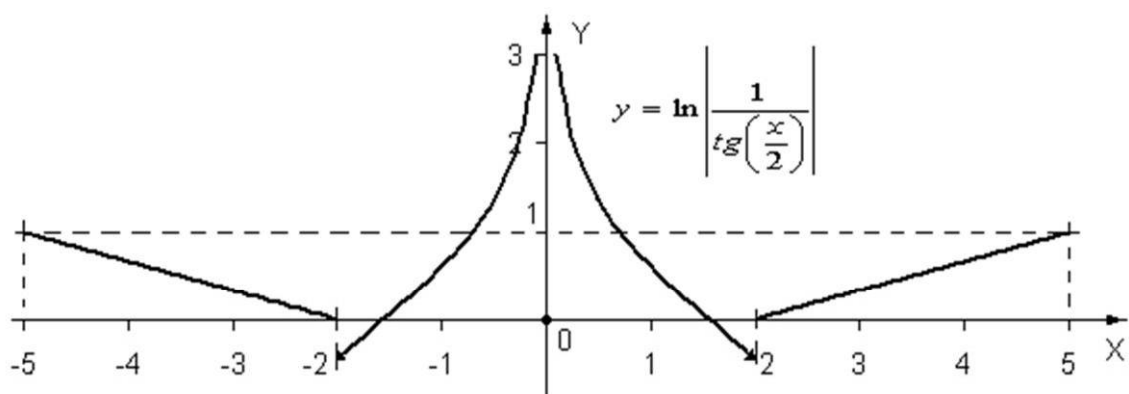
22)



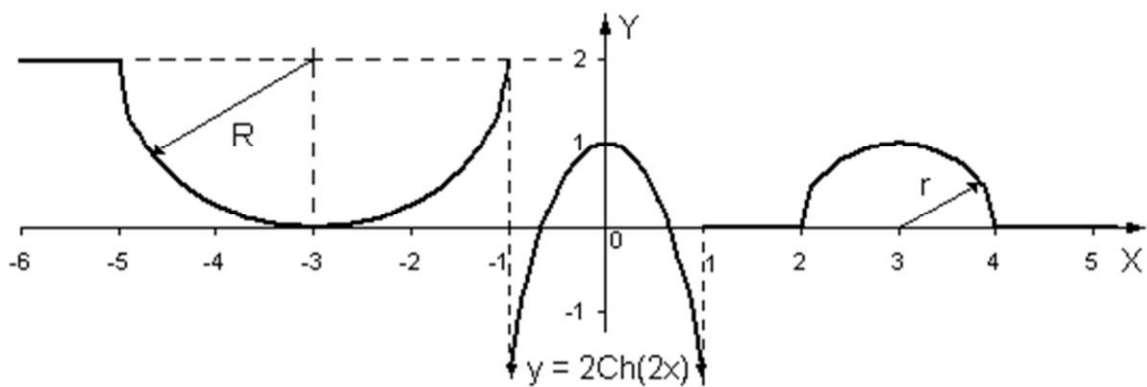
23)



24)

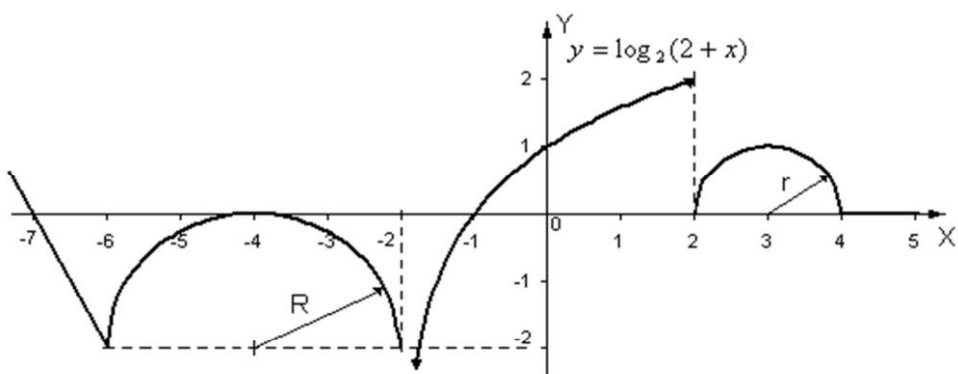


25)

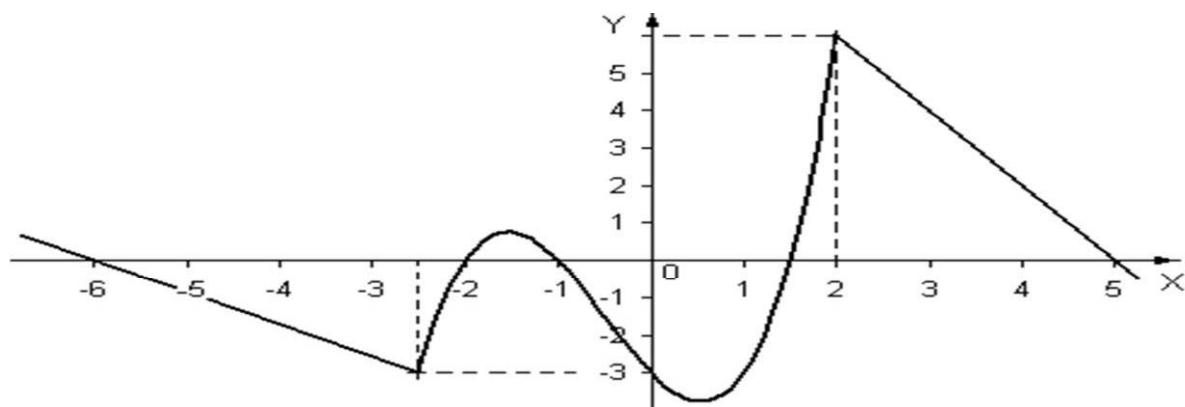


Гиперболический косинус может быть вычислен по формуле: $\text{Ch}(x) = \frac{1}{2}(e^x + e^{-x})$.

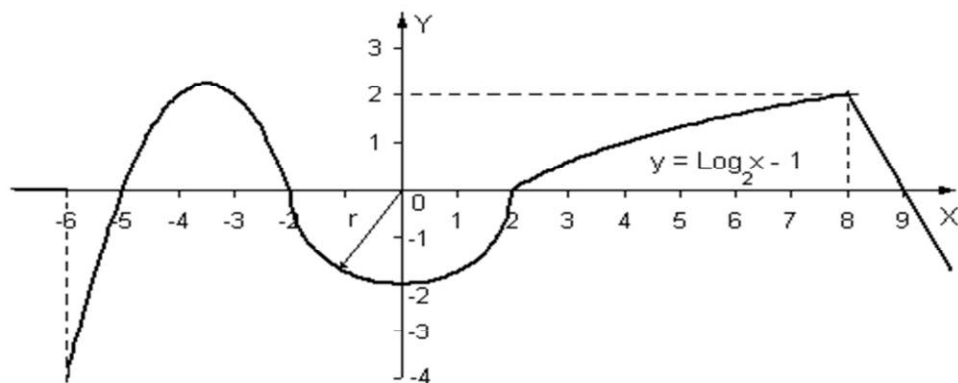
26)



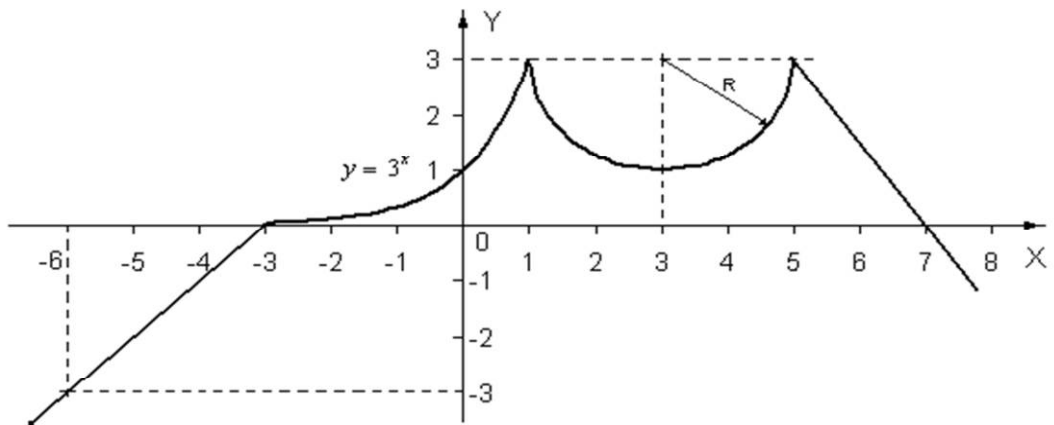
27)



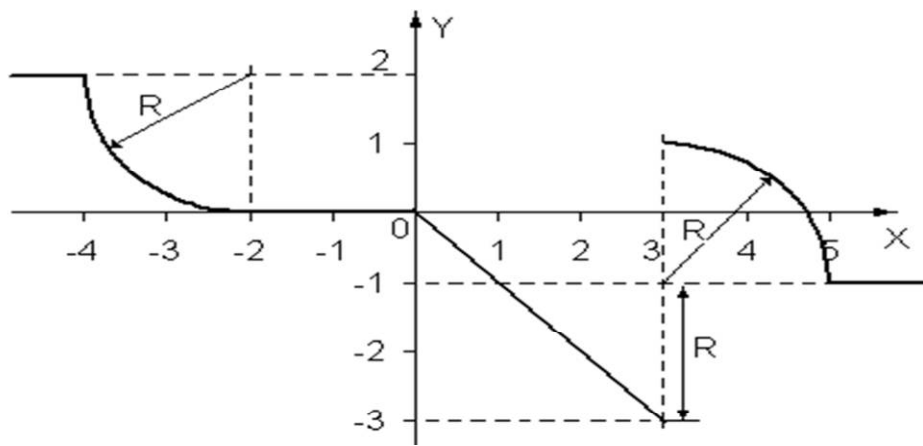
28)



29)



30)

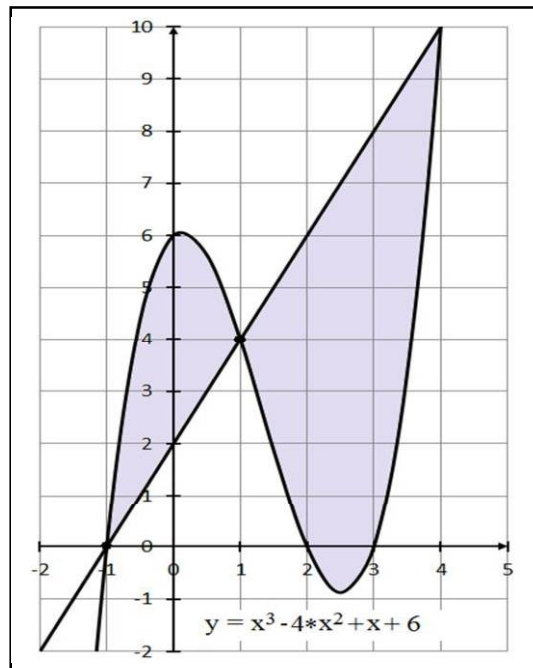


Дополнительное требование к заданию №30: Программа должна быть написана так, что бы решения получались при различных значениях R , вводимых с клавиатуры. Центр положения левой четверти окружности изменяется в соответствии с введённым радиусом, а правой - остаётся постоянным при $X = 3, Y = -1$.

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 четырем).

Нам важно 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 уравнение прямой (уравнение кривой приведено на рисунке). Из рисунка получаем координаты двух точек, через которые проходит прямая (-1, 0) и (1, 4). Уравнение прямой линии в общем виде: $y=kx+b$. Подставим значения выбранных точек и вычислим параметры k и b. В результате вычислений найдем уравнение прямой: $y = 2x+2$.

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, введенные 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 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 в область")
else:
    print("не попадает в область")
```

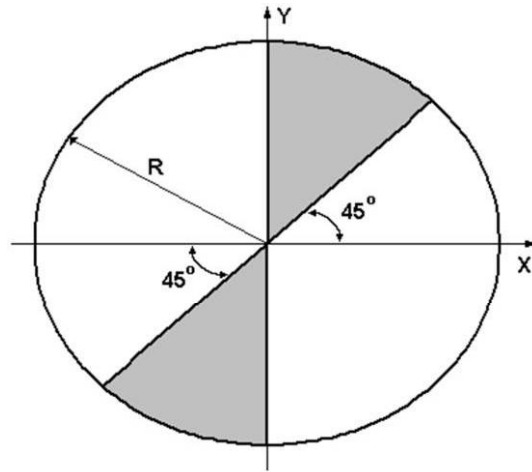
Замечание к тексту программы

Для переноса длинных строк в Python используется обратный слэш (\) за которым сразу следует возврат каретки (Enter)

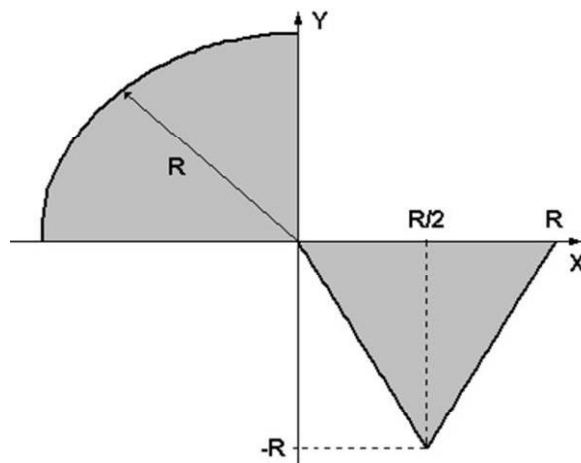
Задание к лабораторной работе №2 "Разветвляющиеся вычислительные процессы". Задание 2

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.

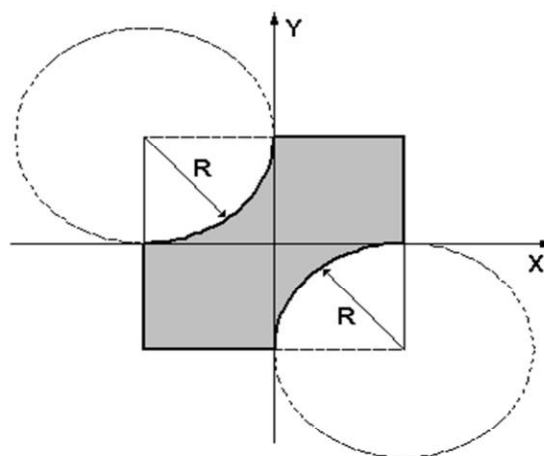
1)



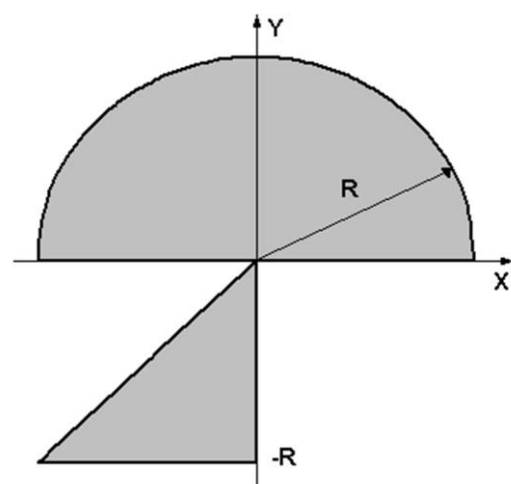
2)



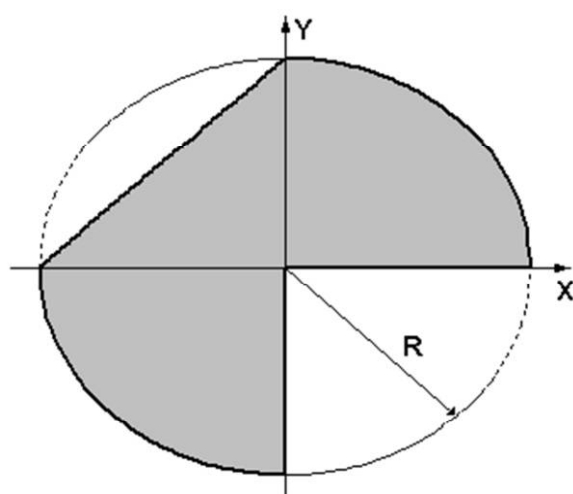
3)



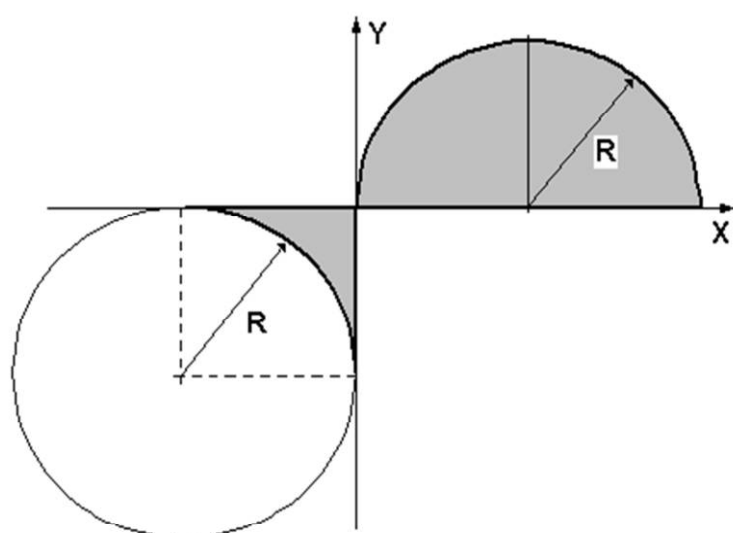
4)



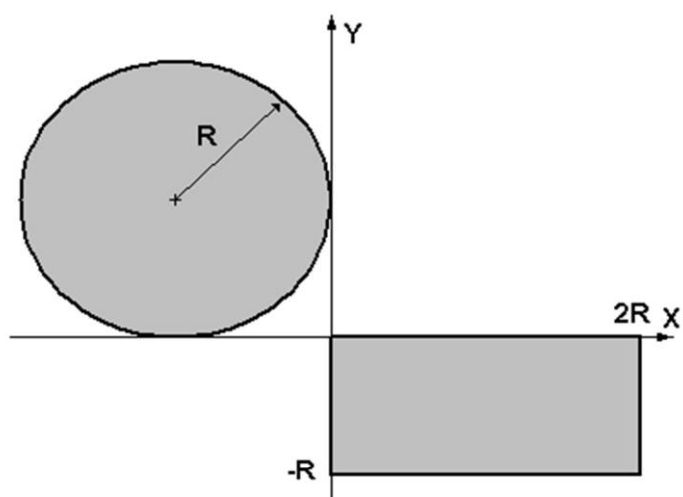
5)



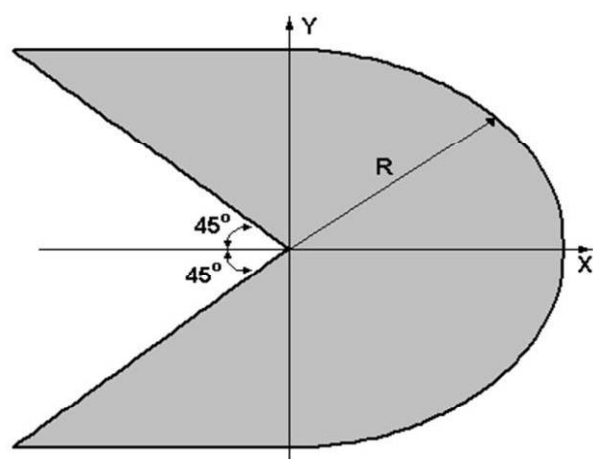
6)



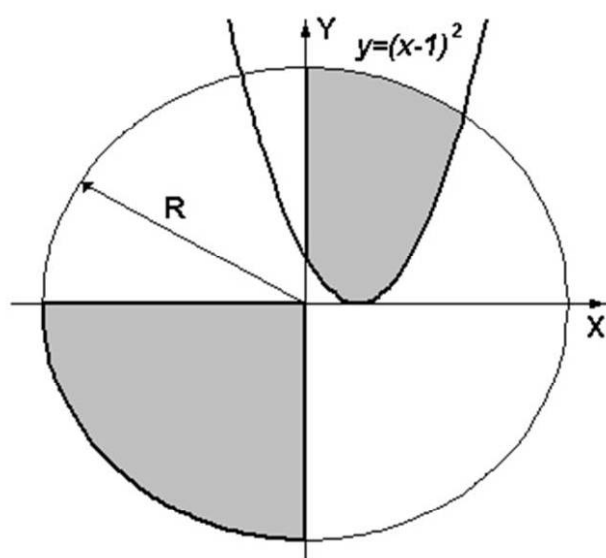
7)



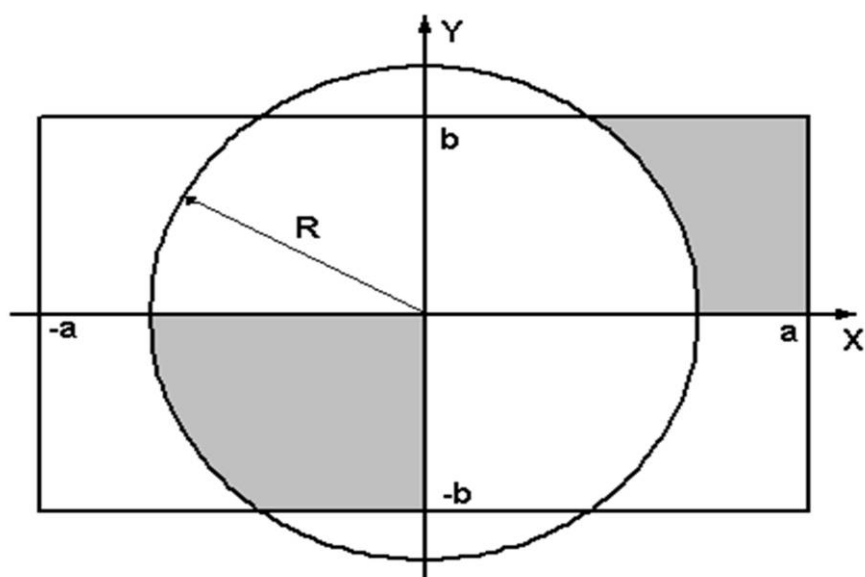
8)



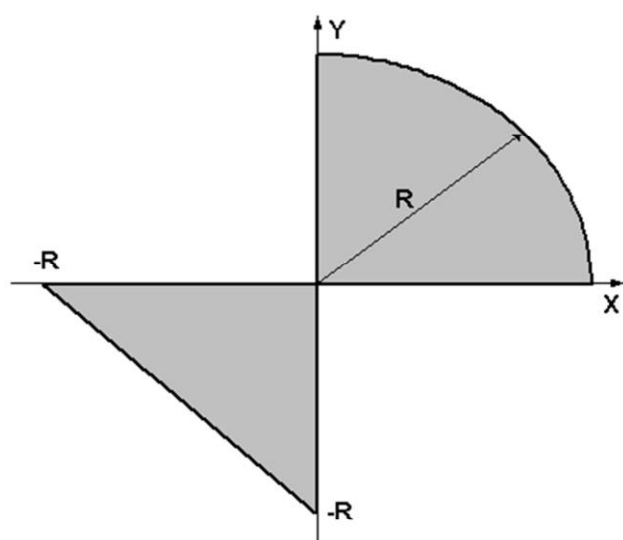
9)



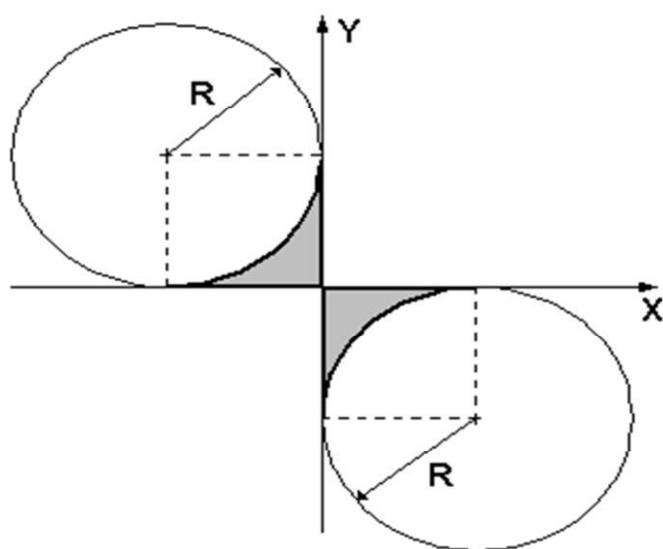
10)



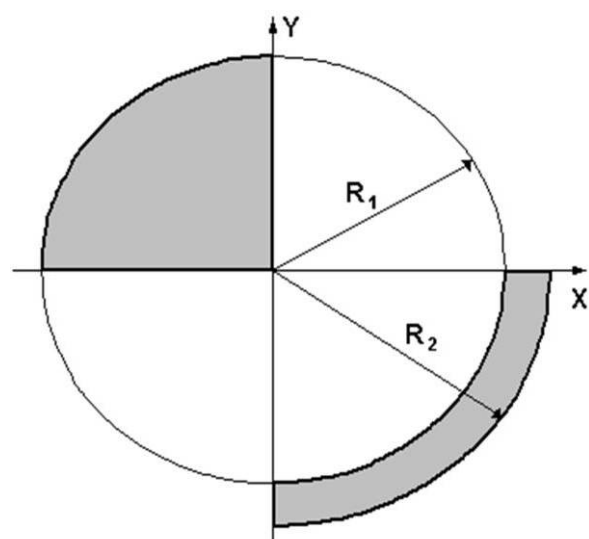
11)



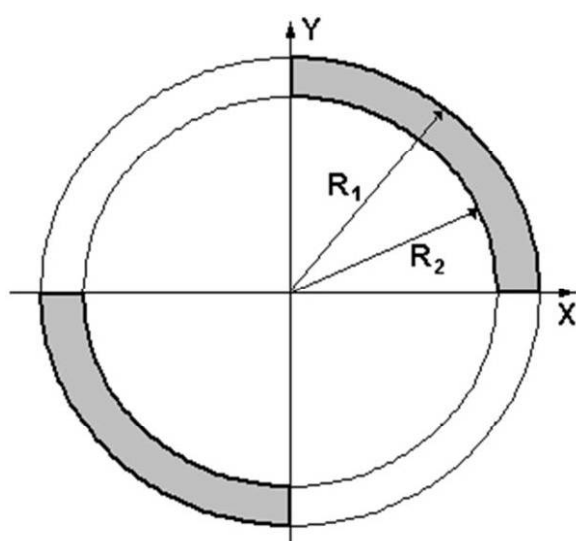
12)



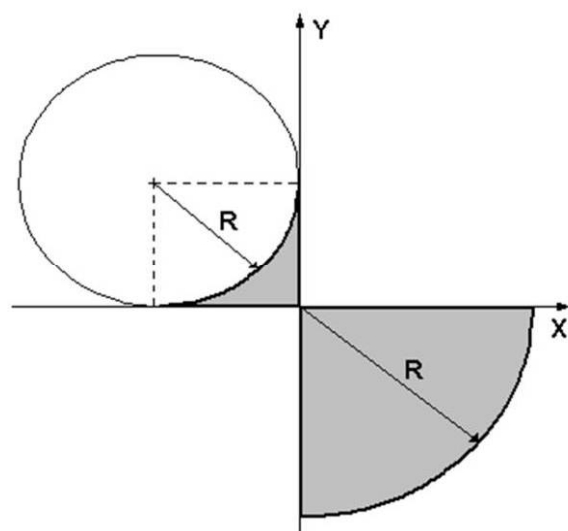
13)



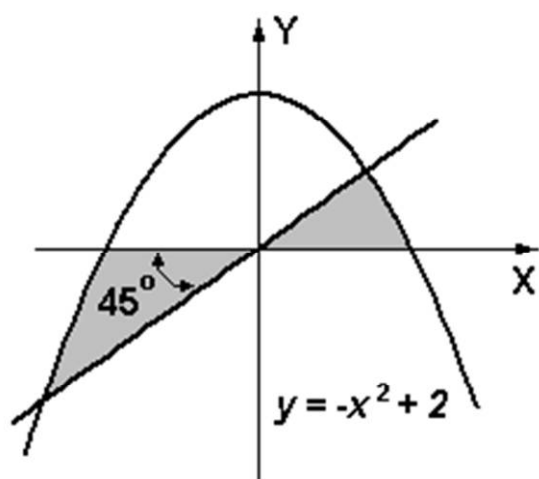
14)



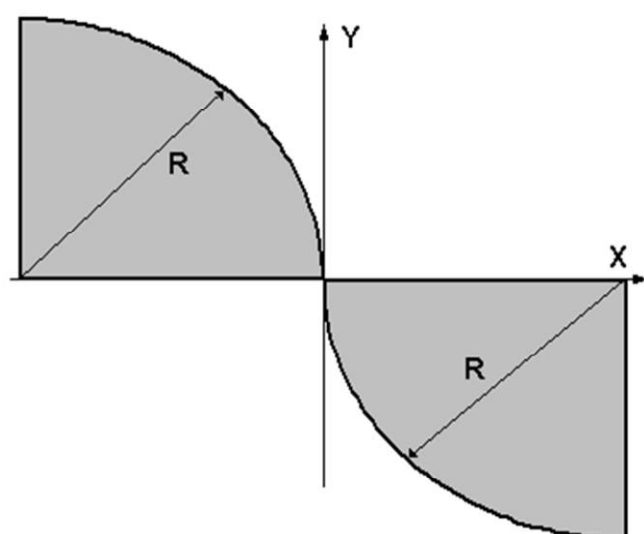
15)



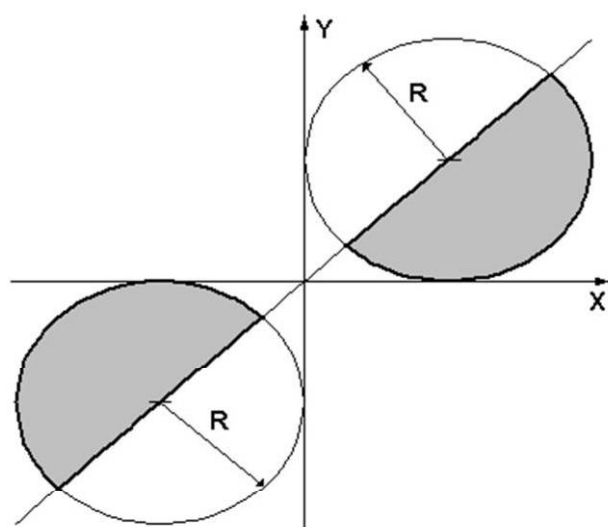
16)



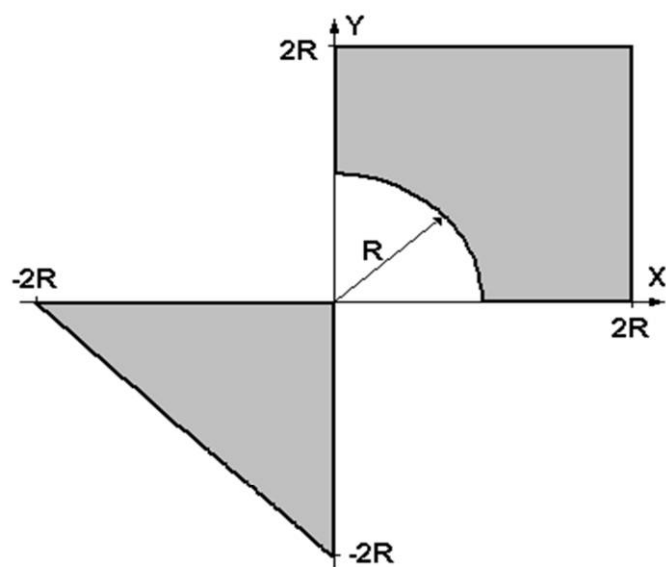
17)



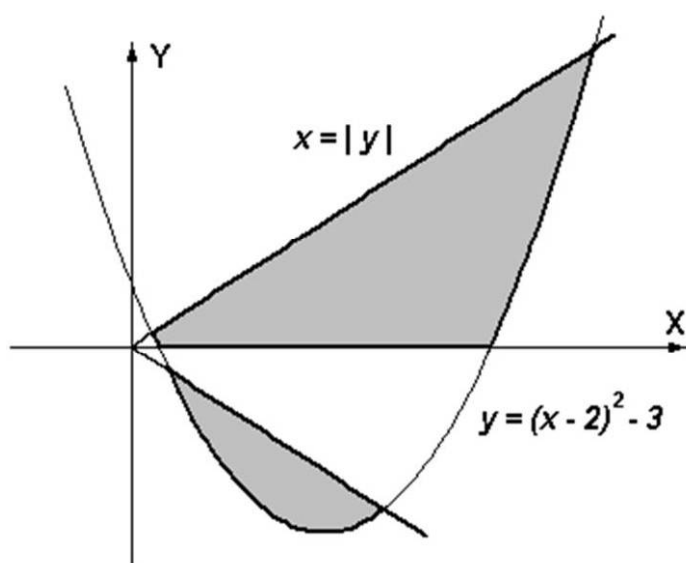
18)



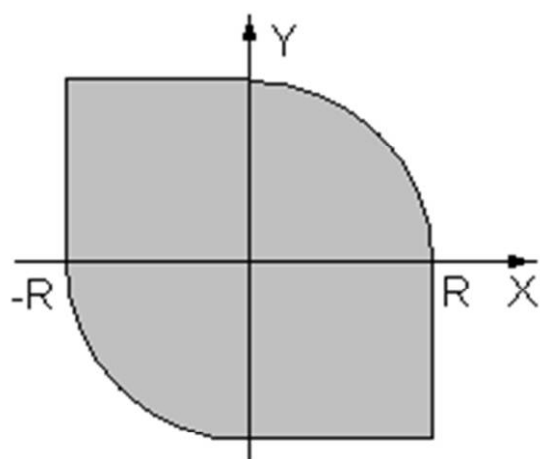
19)



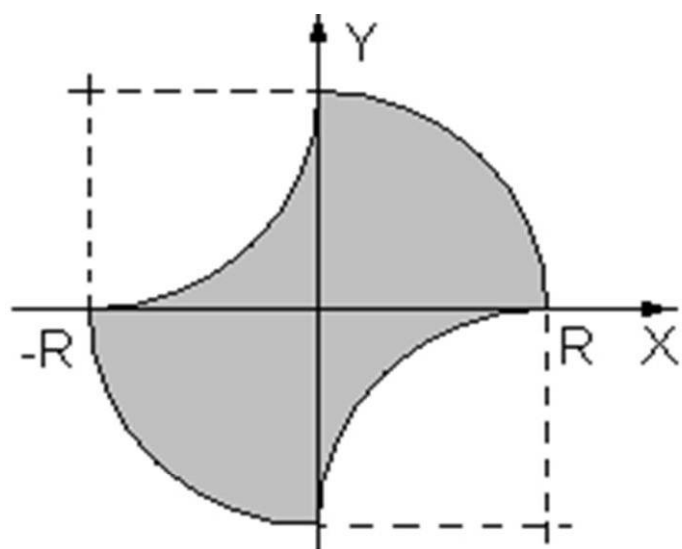
20)



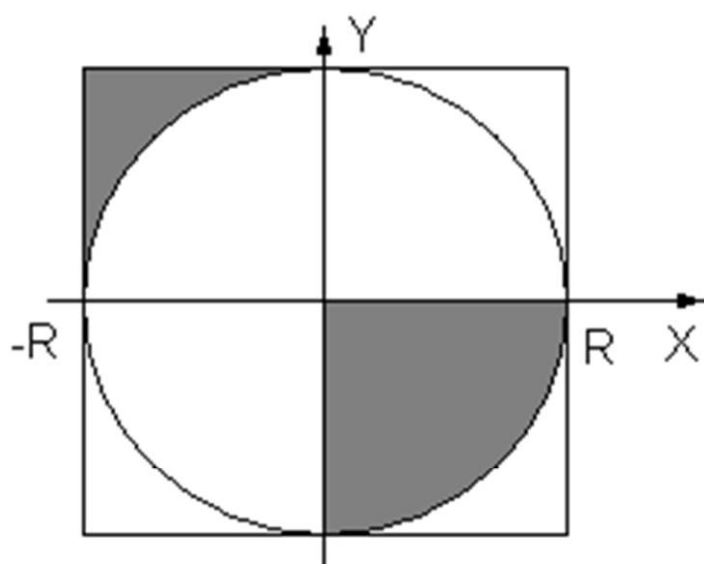
21)



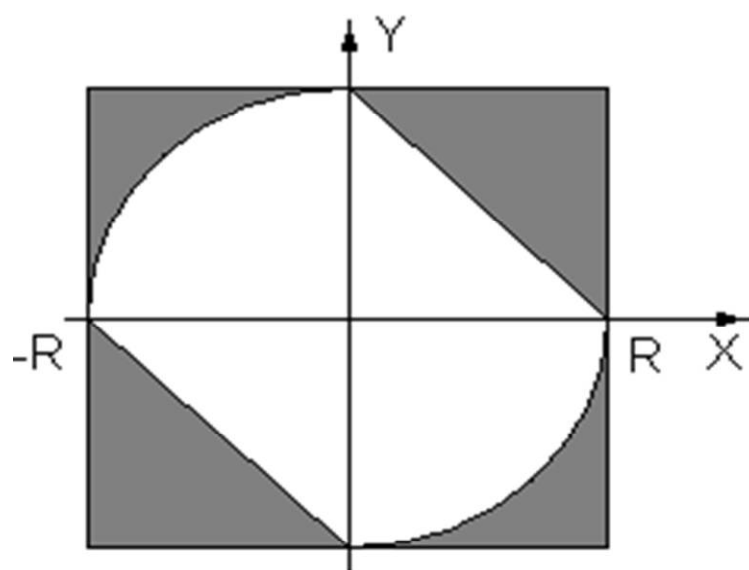
22)



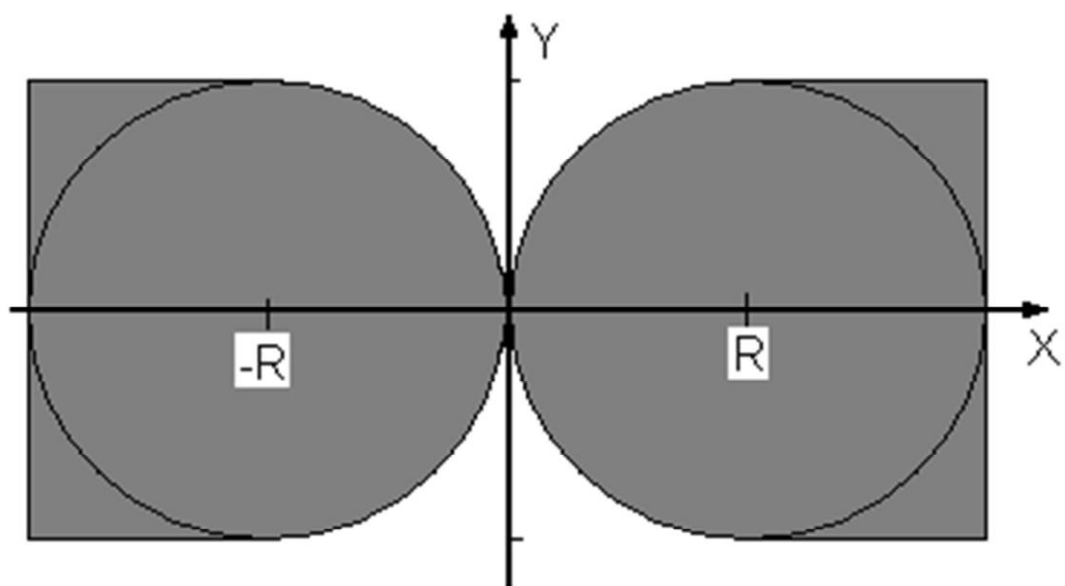
23)



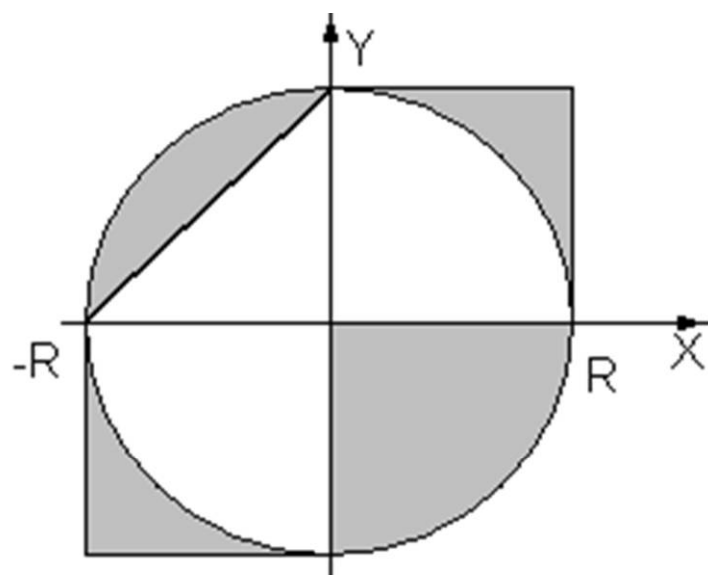
24)



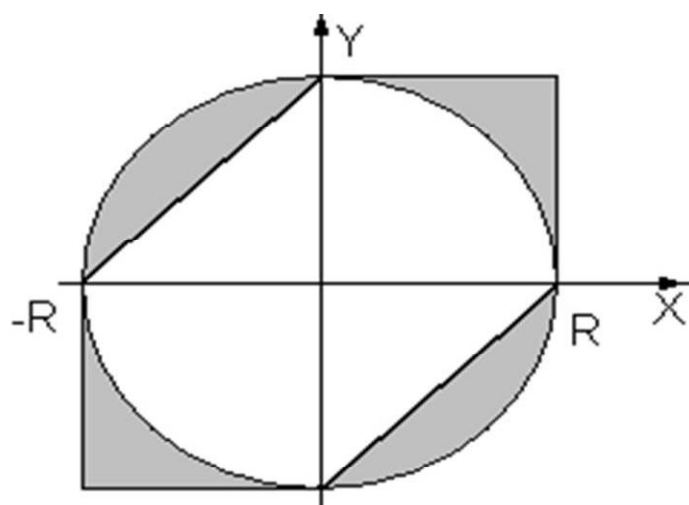
25)



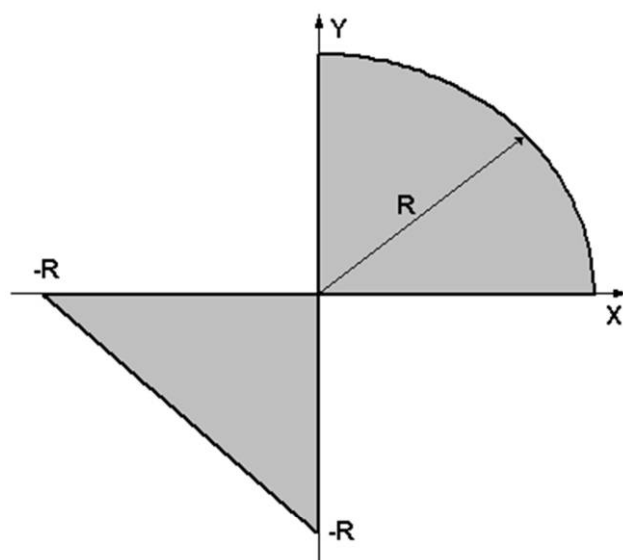
26)



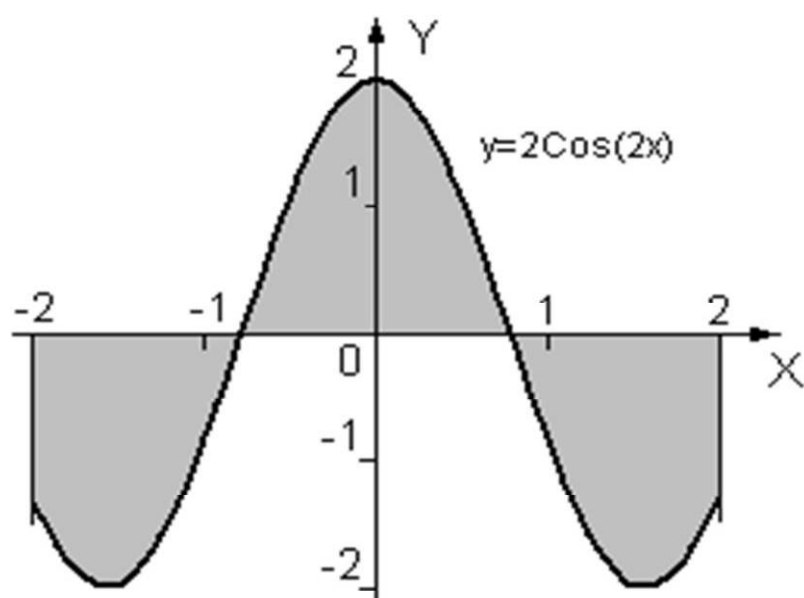
27)



28)



29)



30)

