

Лабораторная работа №2.

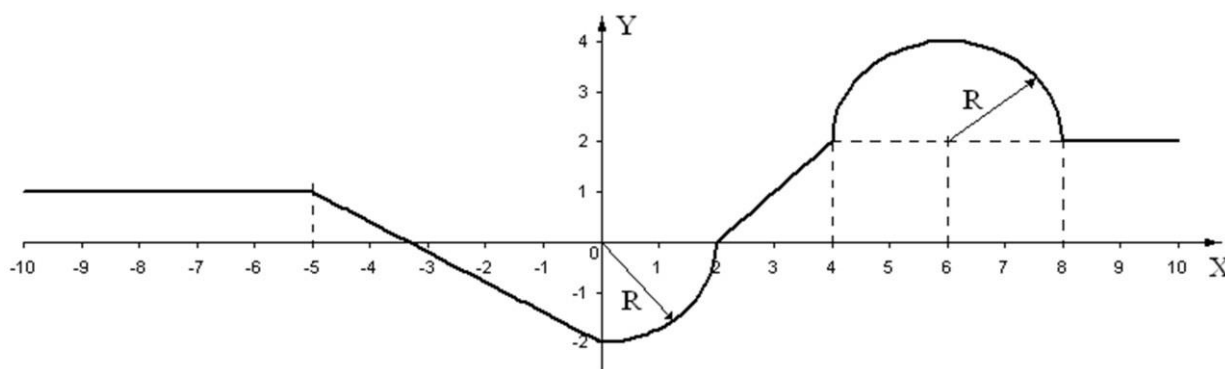
Изучение разветвляющихся алгоритмов, операторов выбора, программирование разветвляющегося вычислительного процесса "Разветвляющиеся вычислительные процессы"

Цели и задачи работы: изучение разветвляющихся алгоритмов, операторов выбора, программирование разветвляющегося вычислительного процесса.

Задание к работе: Реализовать разветвляющийся вычислительный процесс. Самостоятельно решить задачи в соответствии с индивидуальным вариантом.

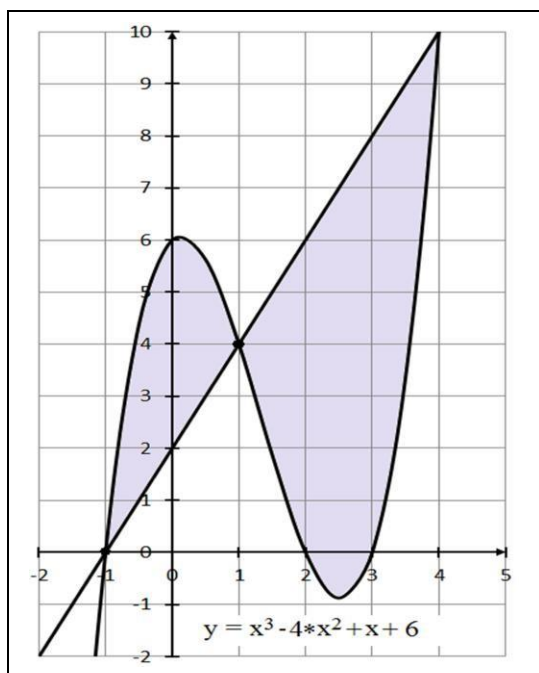
Задание 1. Написать программу, которая по введённому значению аргумента вычисляет значение функции, заданной в виде графика.

Пример:



Задание 2. 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.

Пример:



Теоретическая часть

Когда вы пишете программу, у вас постоянно возникает необходимость выполнить то или иное действие, тот или иной блок кода в зависимости от выполнения условия. Также может возникнуть необходимость какой-то блок кода выполнить несколько раз. Для этого

существуют конструкции управления потоком.

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 четырьмя).

Таким образом, в Python есть возможность использовать выражение if - elif – else. Оператор elif используется тогда, когда нужно проверить друг за другом какие-то условия, не связанные друг с другом.

Например,

```
company = "google.com"  
  
if "my" in company:  
    print("Подстрока my найдена")  
elif "google" in company:  
    print("Подстрока google найдена")  
else:  
    print("Подстрока не найдена")
```

Подстрока google найдена

В Python возможно использовать тернарные операции аналогично языкам C, C++:

```
score_1 = 5  
score_2 = 0  
  
winner = "Argentina" if score_1 > score_2 else "Jamaica"  
  
print(winner)
```

Argentina

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

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

Пример реализации задания 1.

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

Реализация на языке Python:

```
# -*- 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))
```

Следует отметить, что в такой записи алгоритма проверка выполняется для всех условных операторов, в том числе и тех, которые следуют за вычисленными. 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.

Реализация на языке Python:

```
# -*- 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))
```

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
```

Пример реализации задания 2.

Для решения задачи требуется уравнение прямой (уравнение кривой приведено на рисунке). Из рисунка получаем координаты двух точек, через которые проходит прямая линия (-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: $y = 2x+2$.

Тестовые примеры

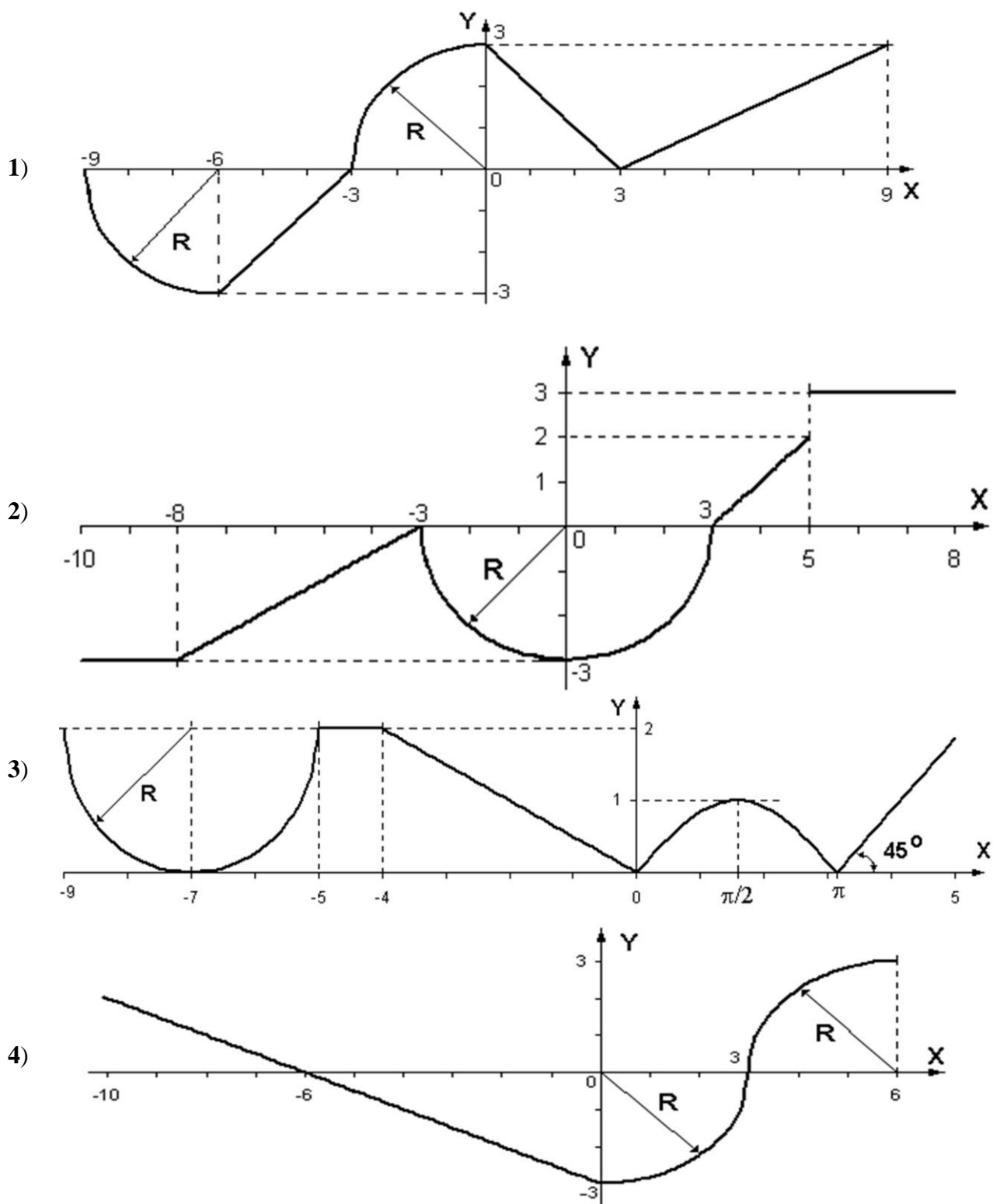
X	Y	Результат
-1	0	Попадает
-0.5	-1	Не попадает
0	3	Попадает
1	4	Попадает
1	5	Не попадает
1.5	1	Не попадает
2	3	Попадает
2.5	-1	Не попадает
2.5	-0.3	Попадает
3.5	10	Не попадает

Листинг программы

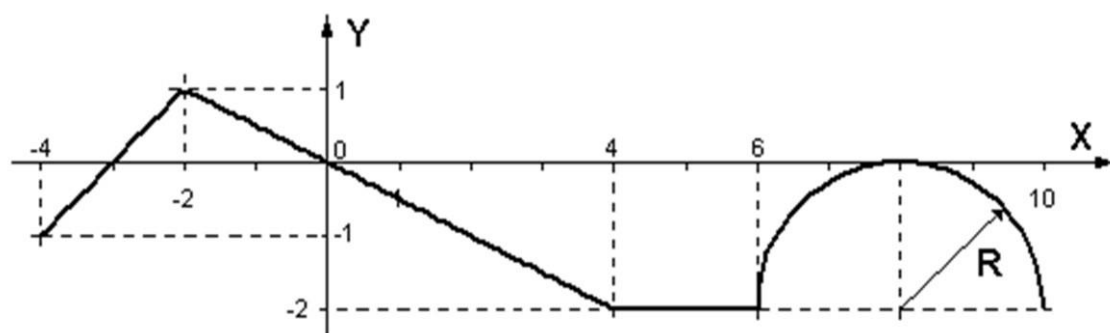
```
# -*- coding: cp1251 -*-
from math import *
flag = 0
print('Введите координаты точки')
x = float(input('X='))
y = float(input('Y='))
if (x < -1) or (x > 4):
    flag = 0      #False
if (x>=-1) and (x<1) and (y>=2*x+2) \
    and (y<=x**3-4*x**2+x+6) or \
    (x>=1) and (x<=4) and (y>=x**3-4*x**2+x+6) \
    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")
```

Задание 2.1

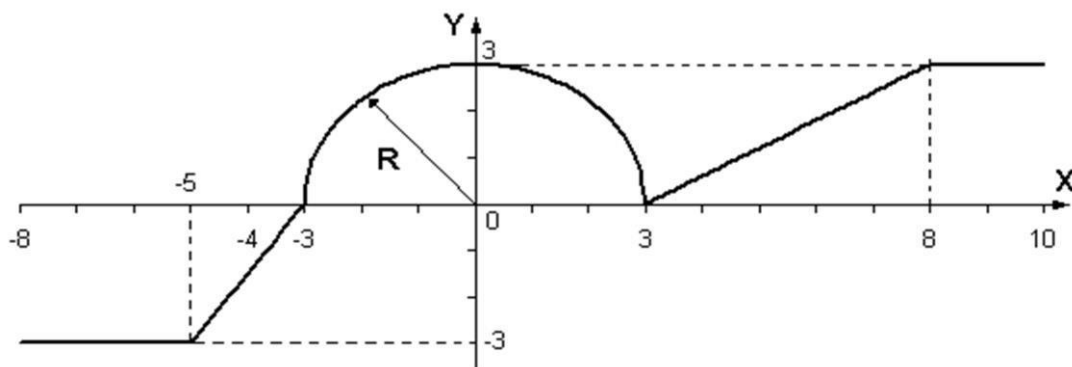
Написать программу, которая по введенному значению аргумента вычисляет значение функции, заданной в виде графика. Параметры, необходимые для решения задания следует получить из графика и определить в программе.



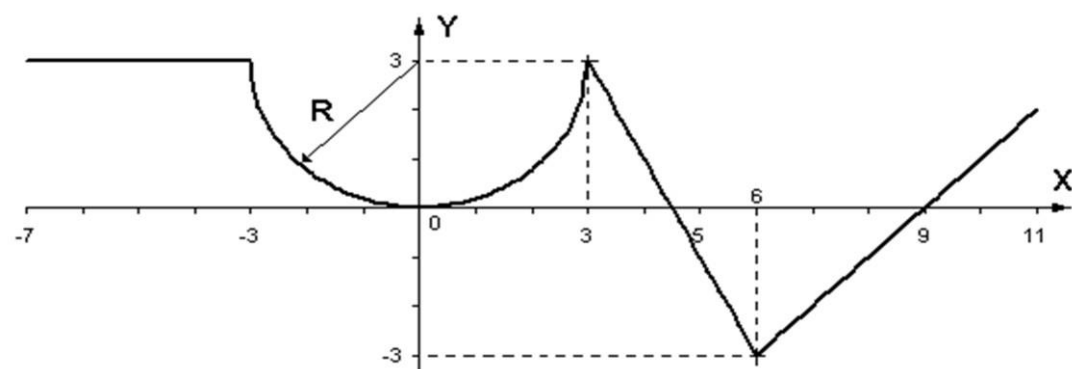
5)



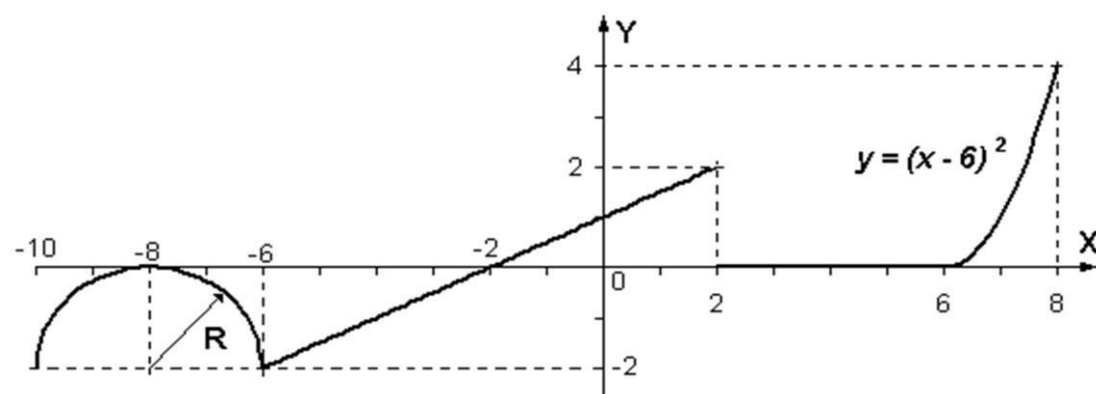
6)



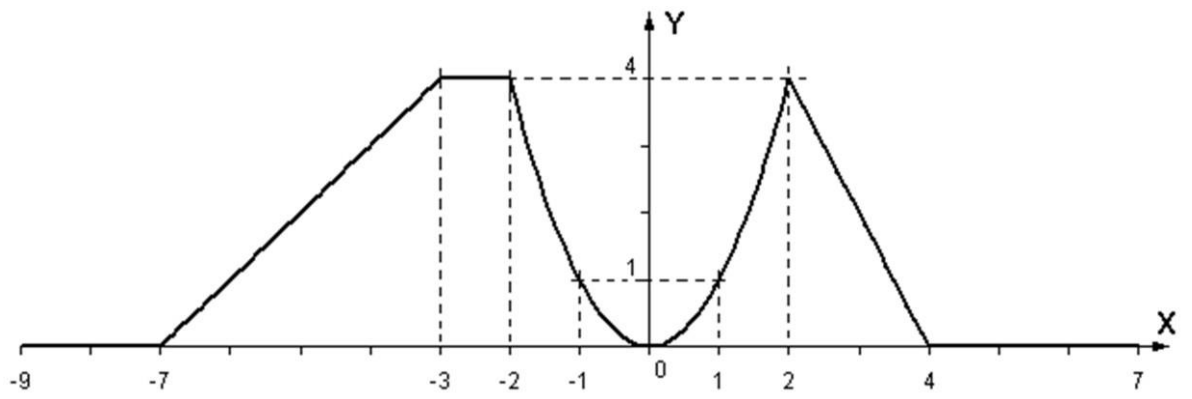
7)



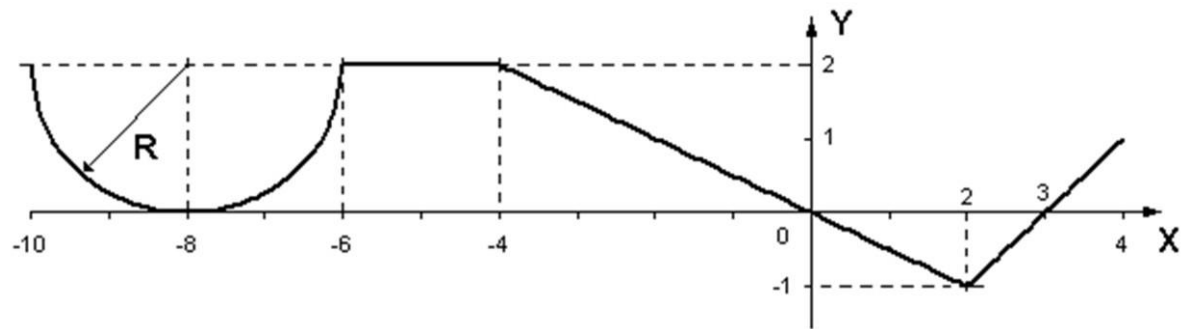
8)



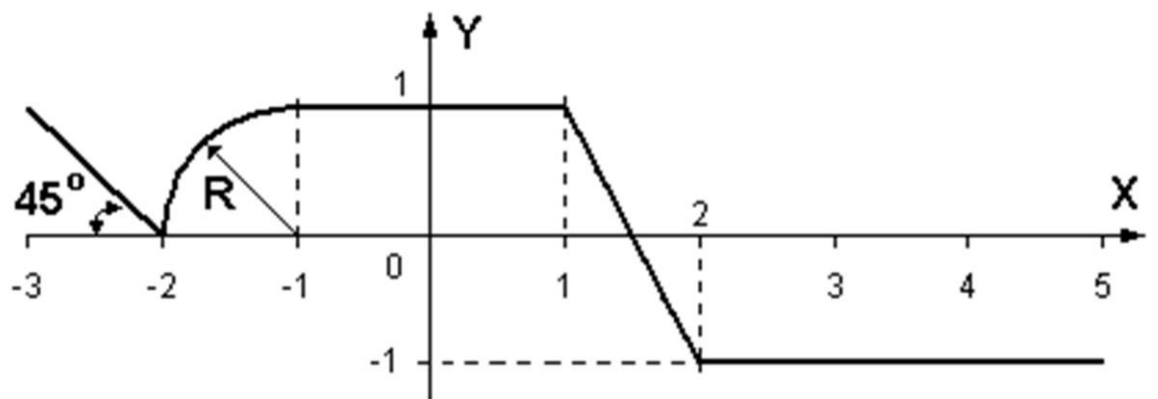
9)



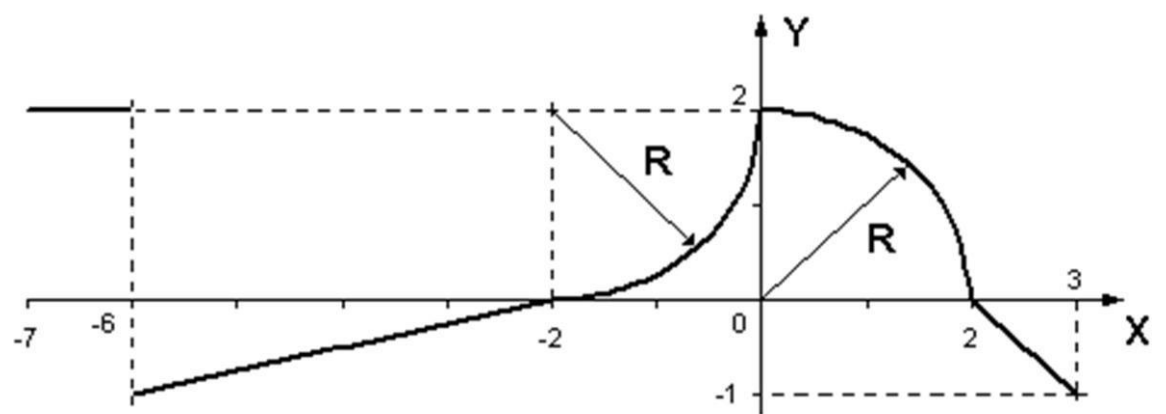
10)



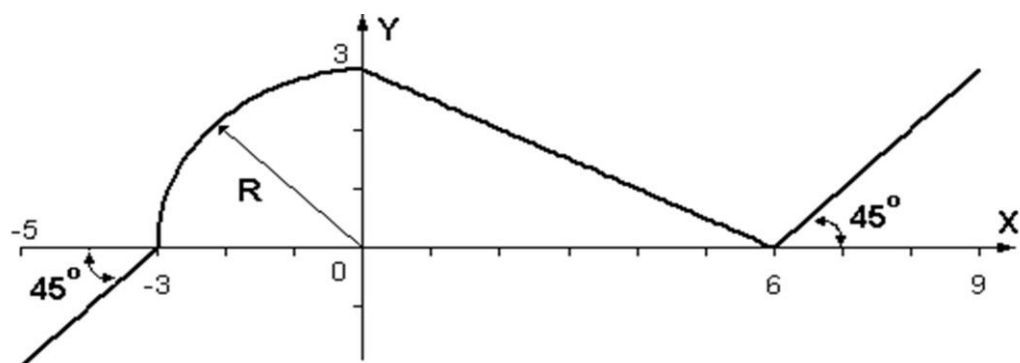
11)



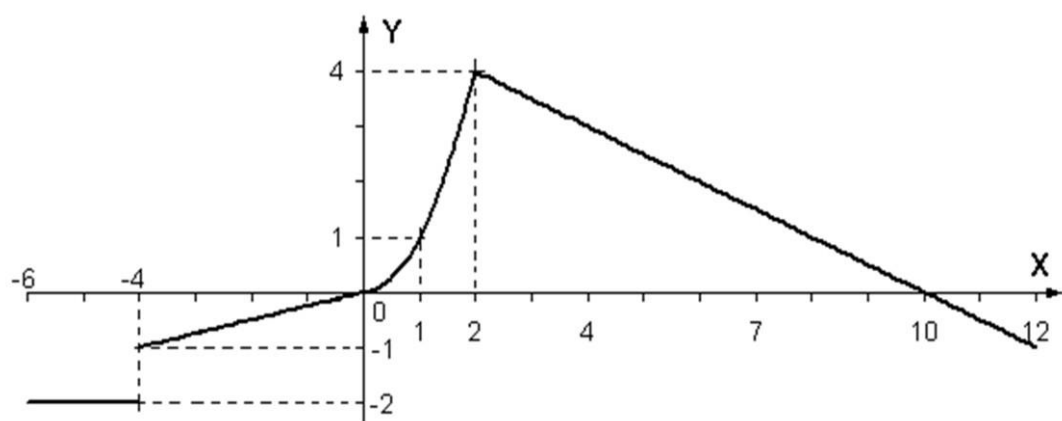
12)



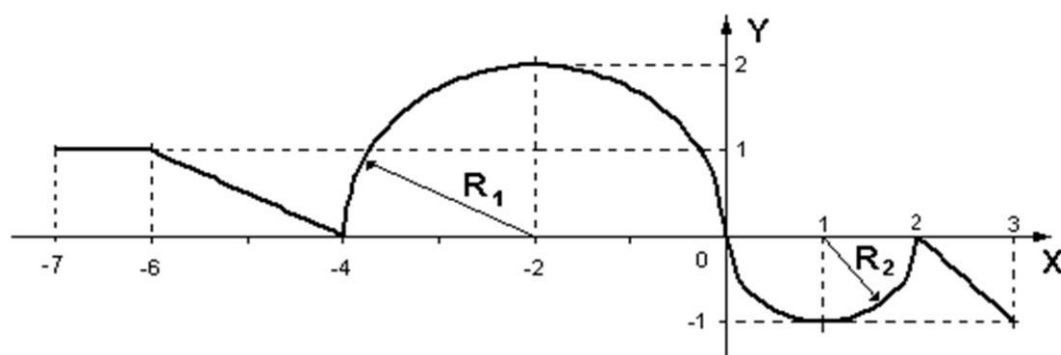
13)



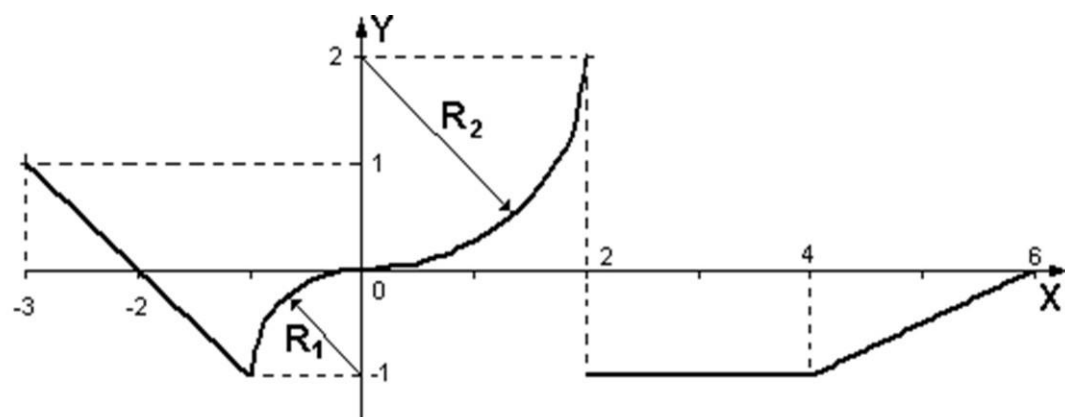
14)



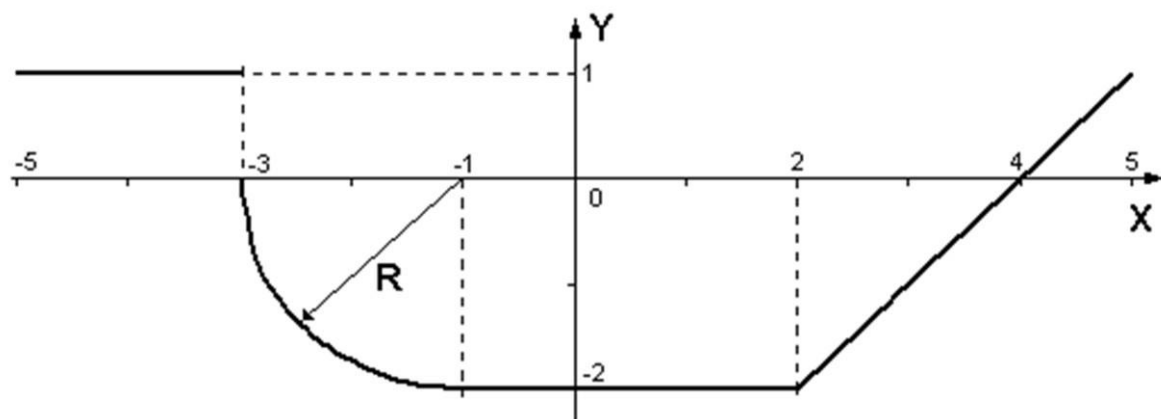
15)



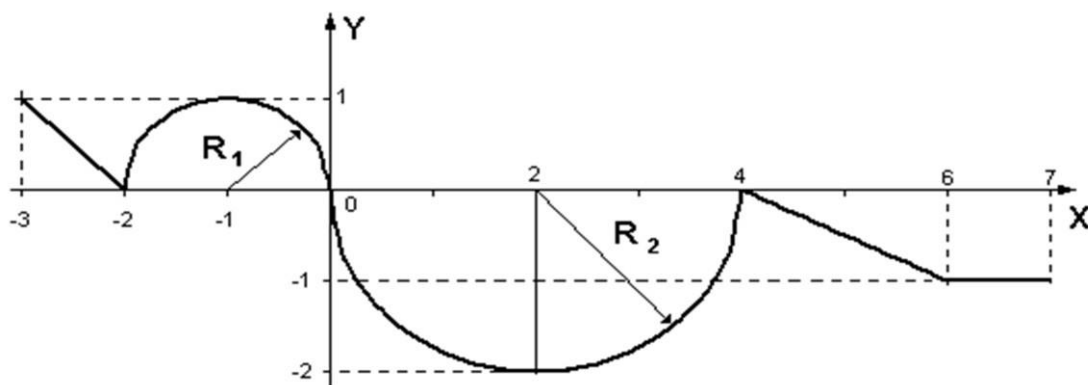
16)



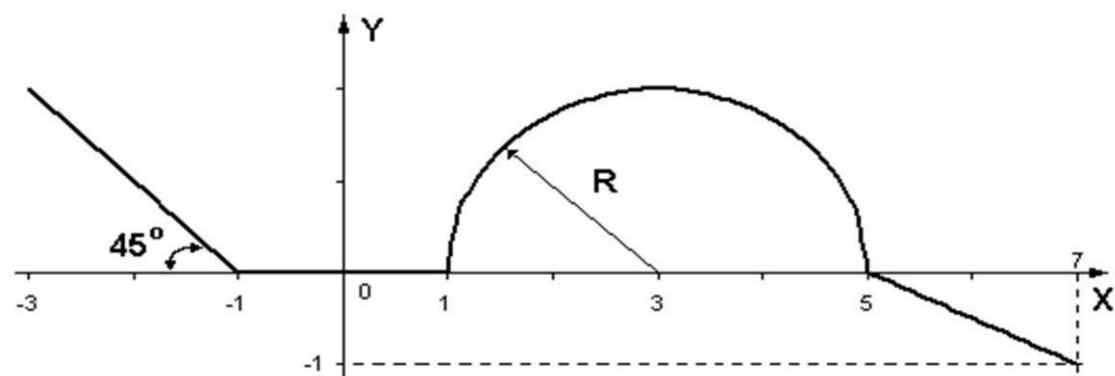
17)



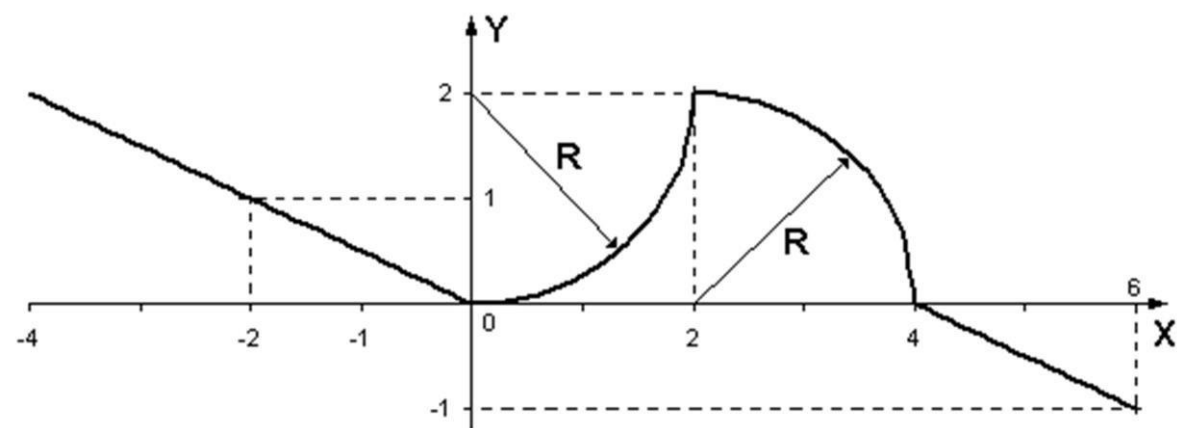
18)



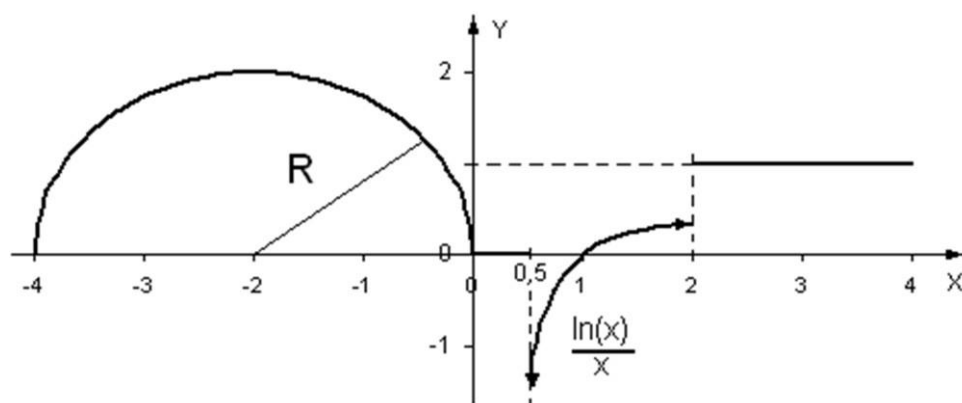
19)



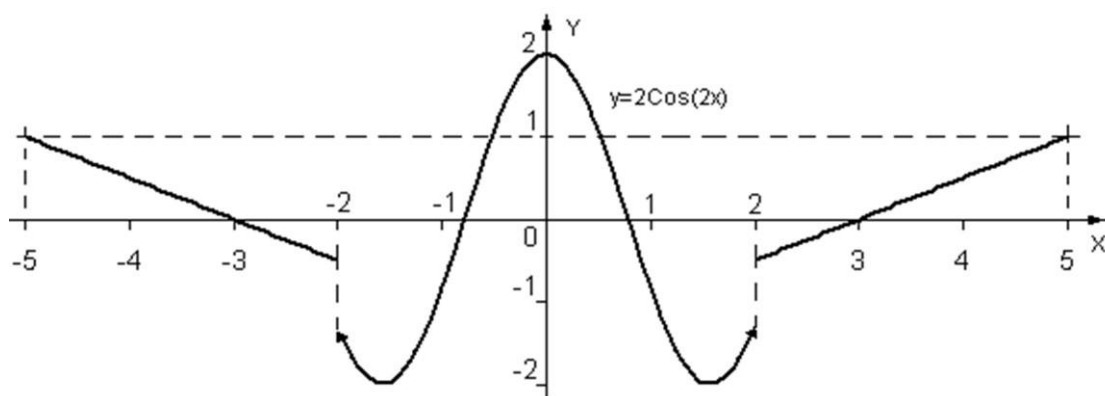
20)



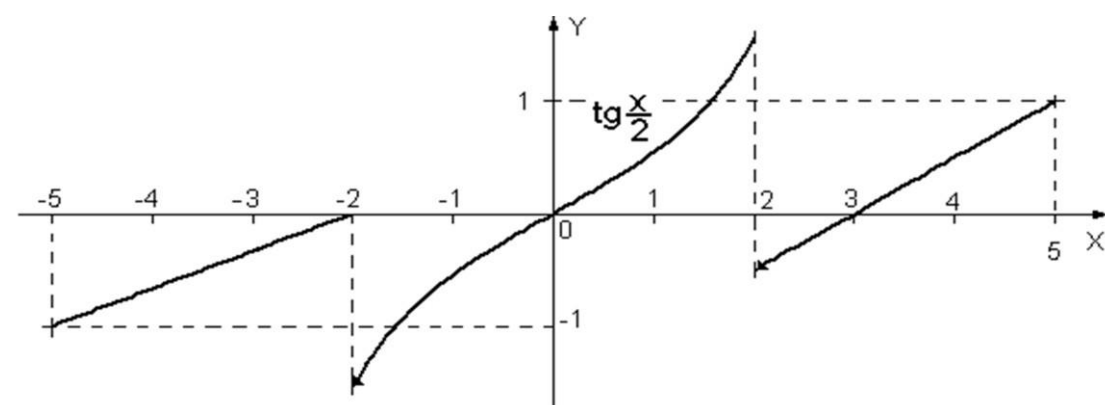
21)



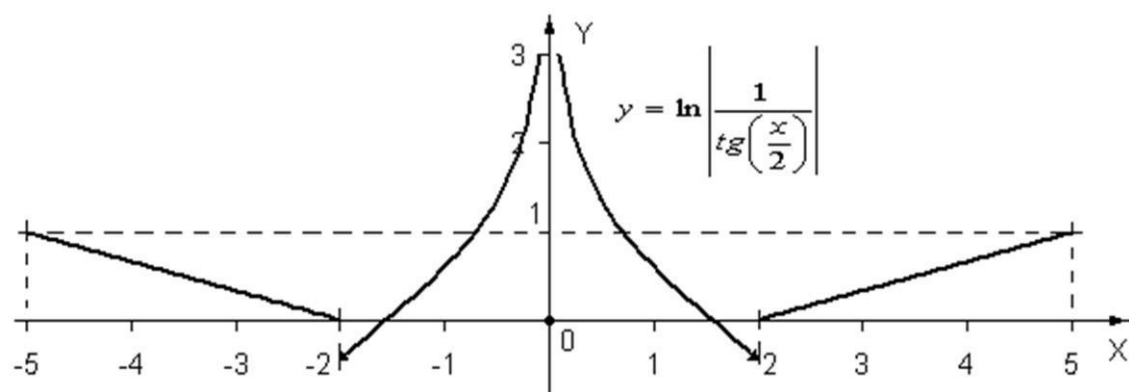
22)



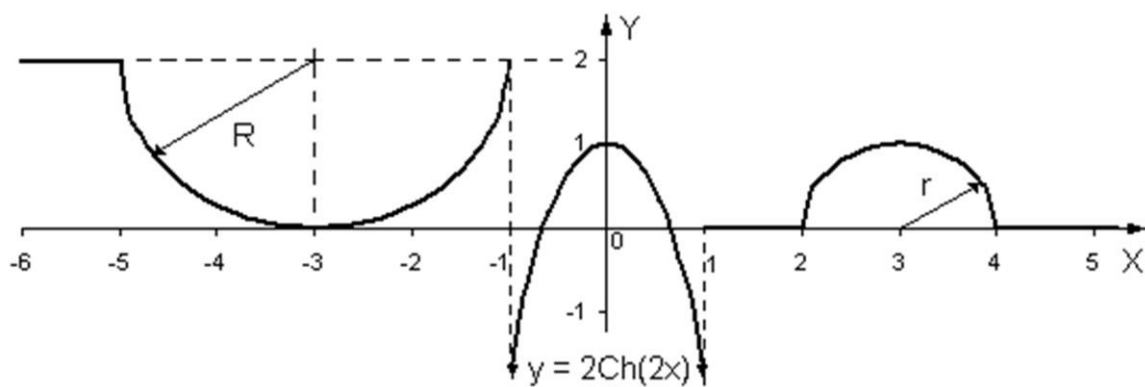
23)



24)

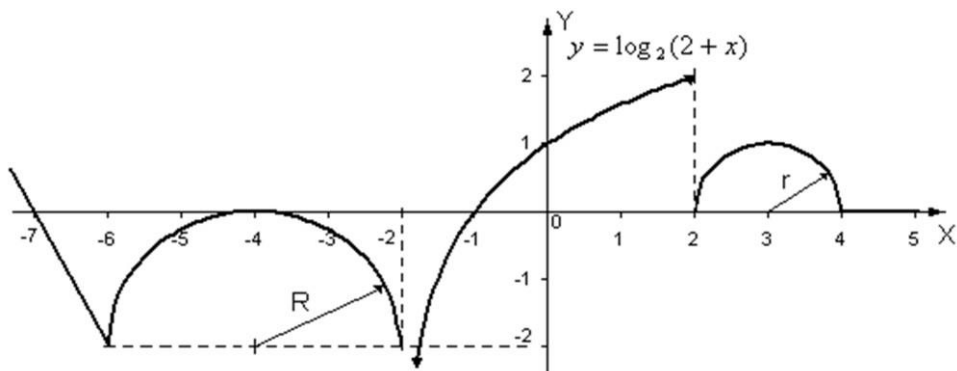


25)

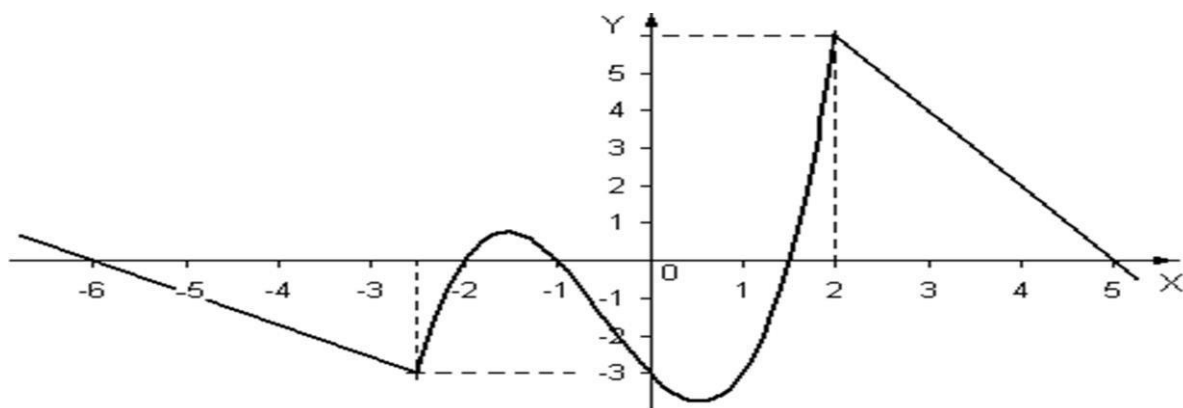


Гиперболический косинус может быть вычислен по формуле: $Ch(x) = \frac{1}{2}(e^x + e^{-x})$.

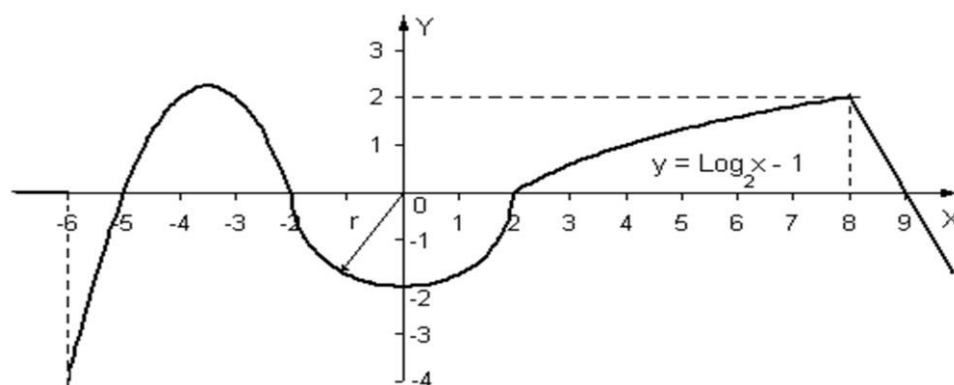
26)



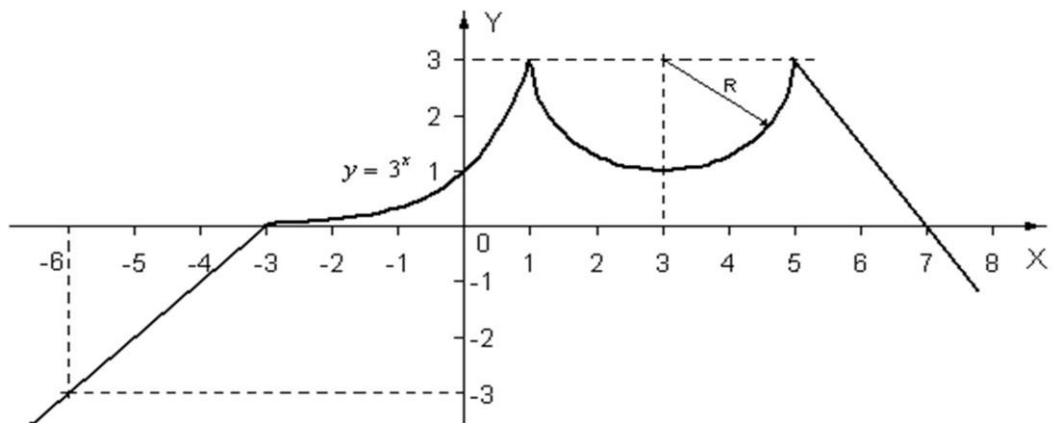
27)



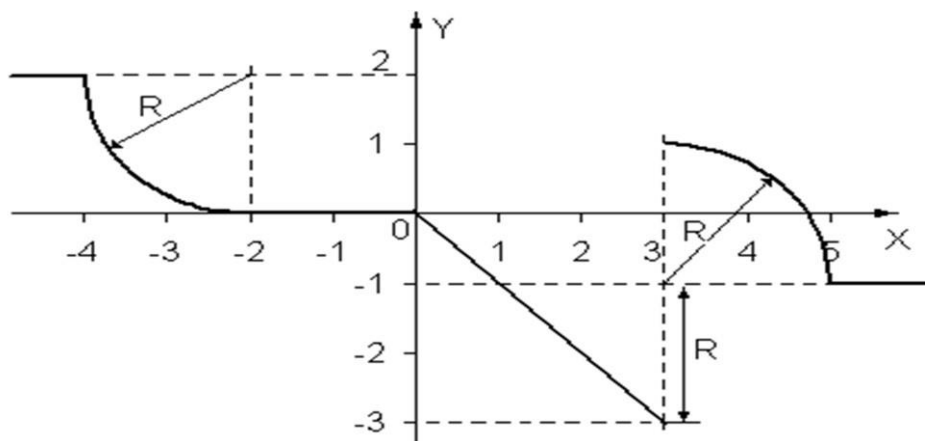
28)



29)



30)

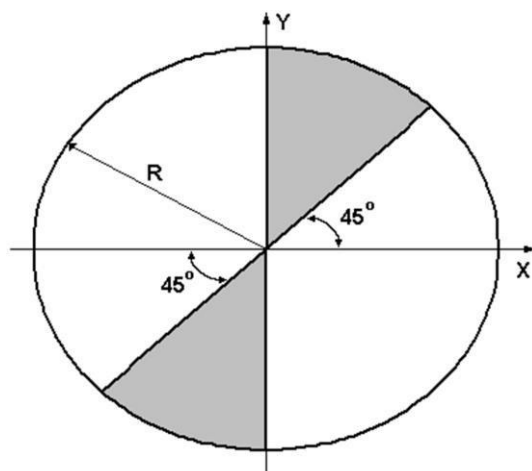


Дополнительное требование к заданию №30: Программа должна быть написана так, чтобы решения получались при различных значениях R , вводимых с клавиатуры. Центр положения левой четверти окружности изменяется в соответствии с введённым радиусом, а правой - остаётся постоянным при $X = 3$, $Y = -1$.

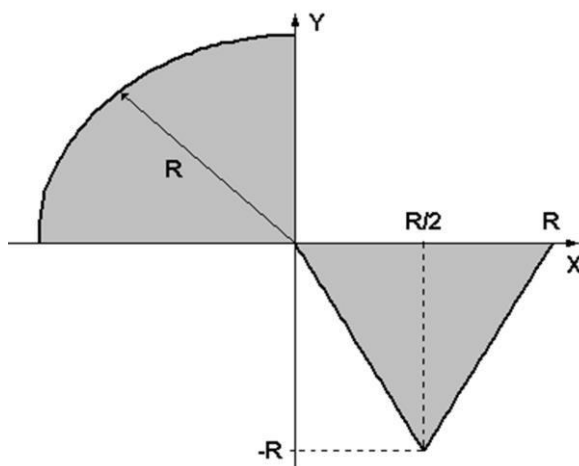
Задание 2.2

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.

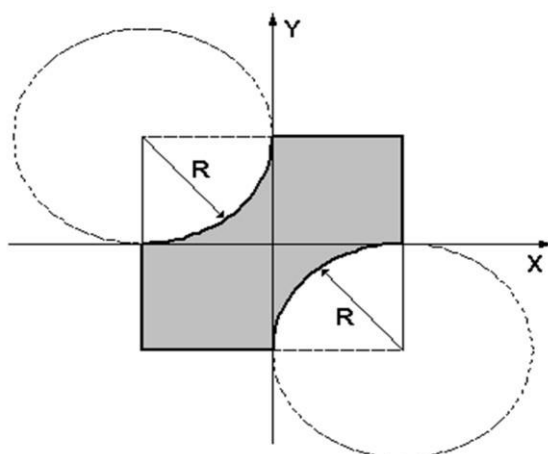
1)



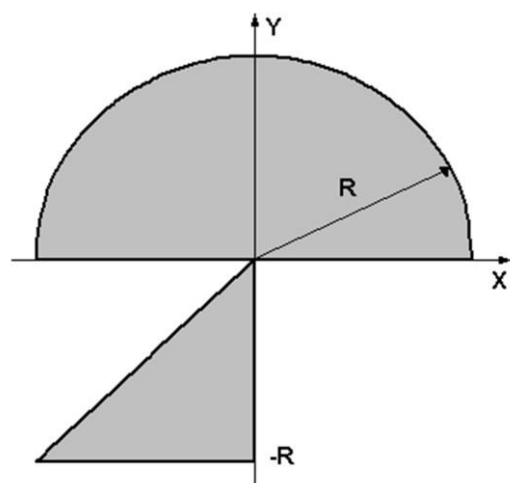
2)



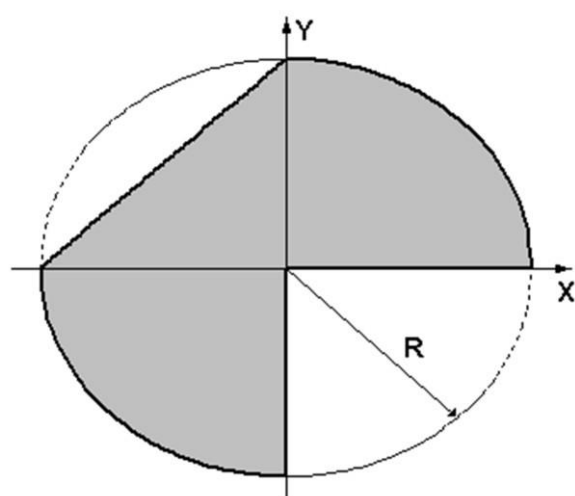
3)



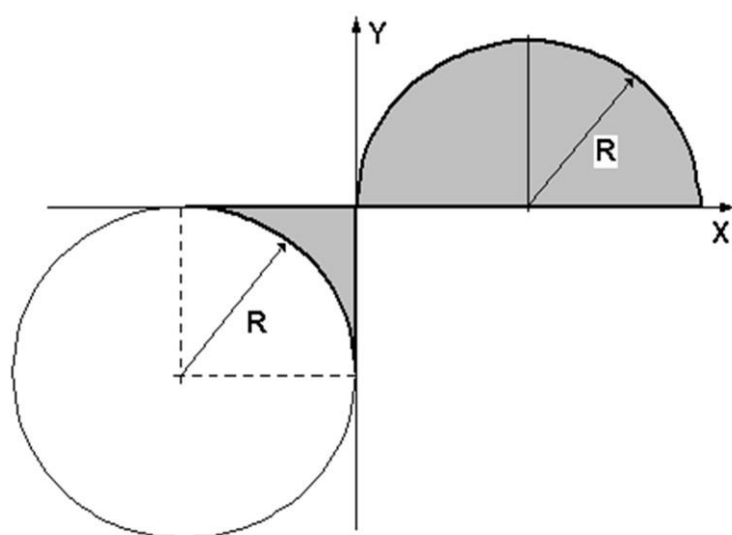
4)



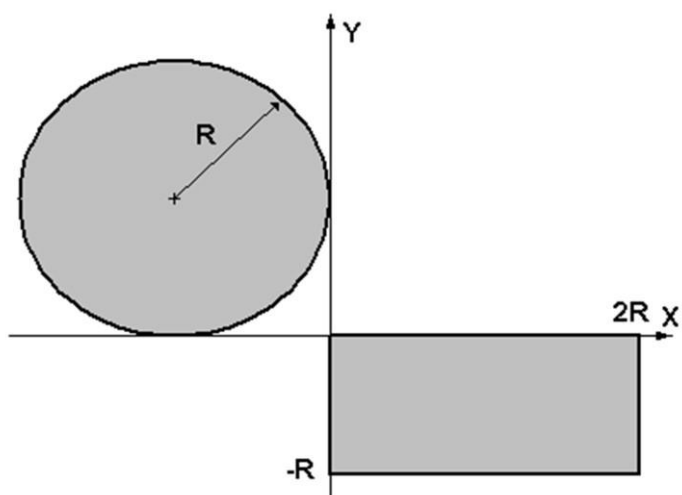
5)



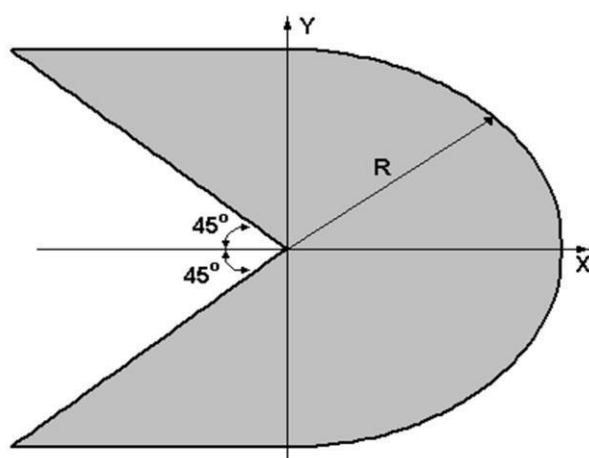
6)



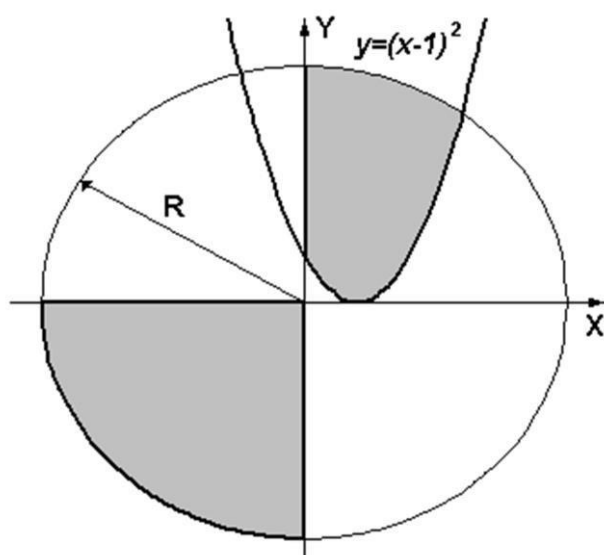
7)



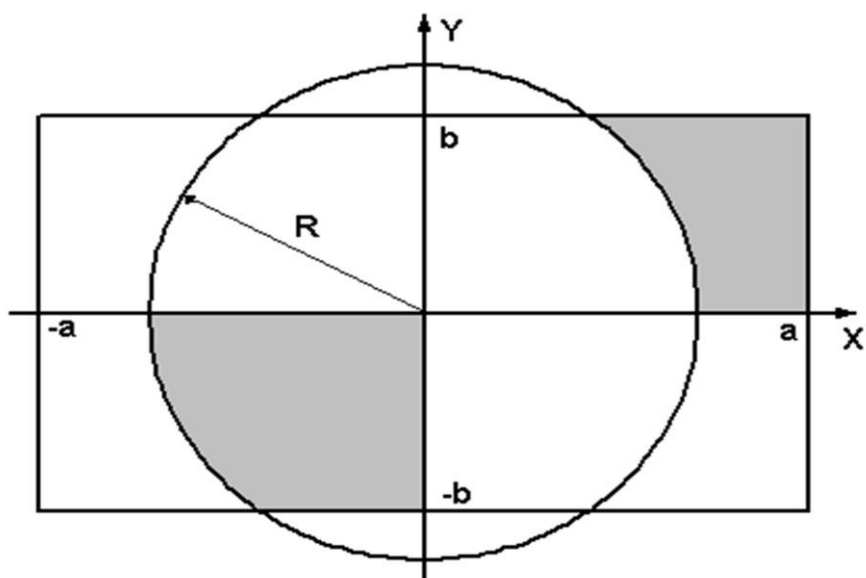
8)



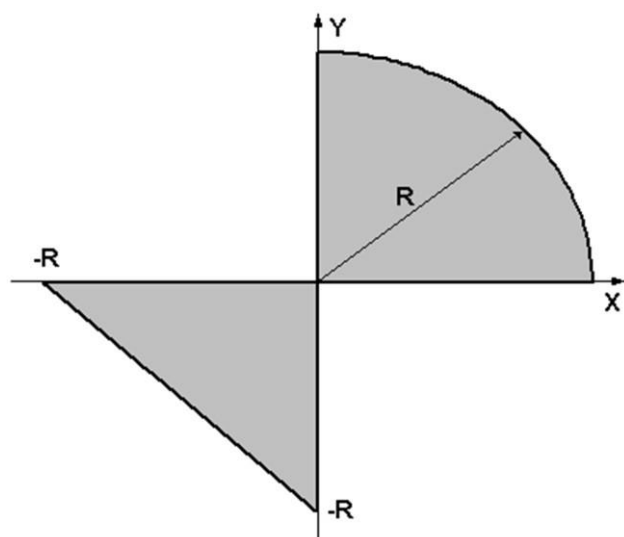
9)



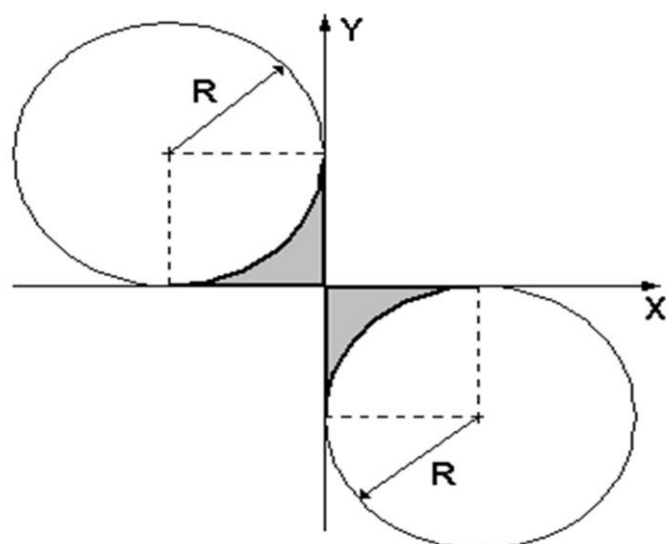
10)



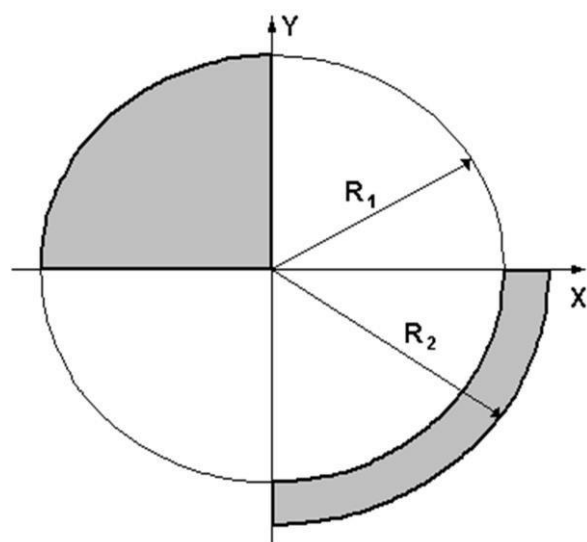
11)



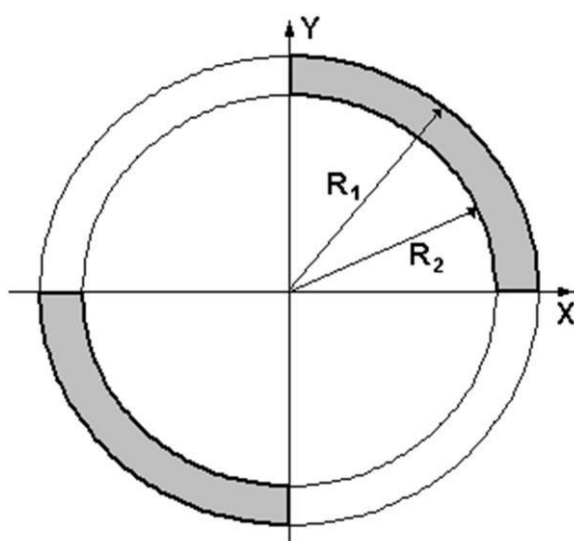
12)



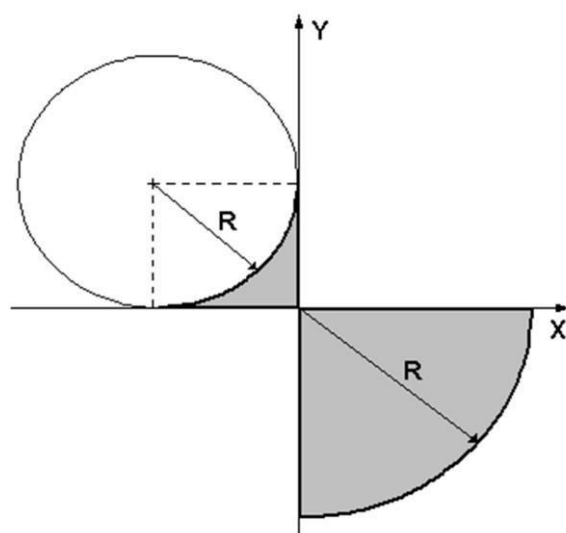
13)



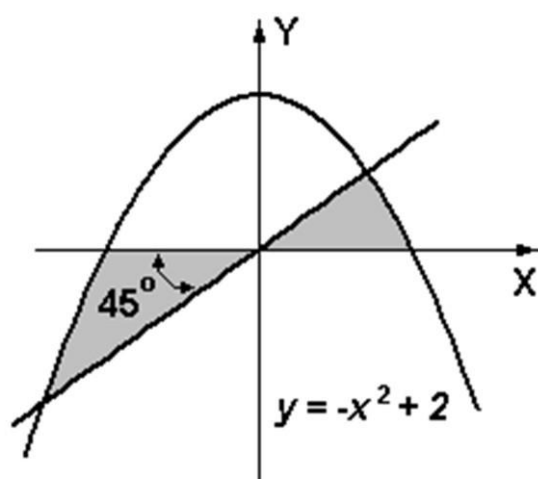
14)



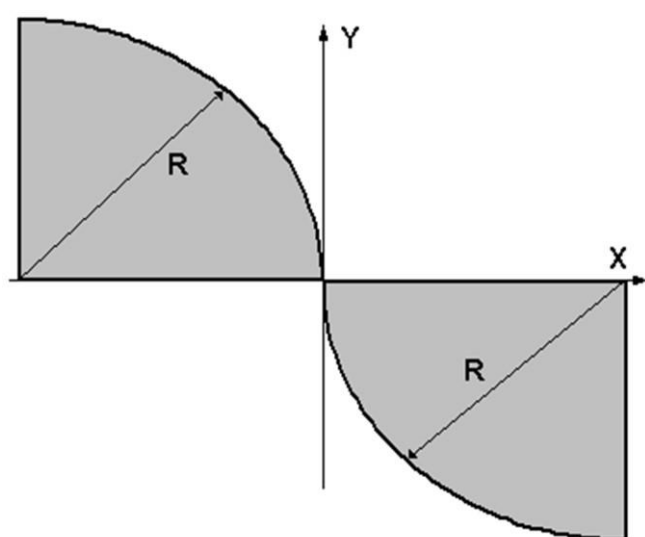
15)



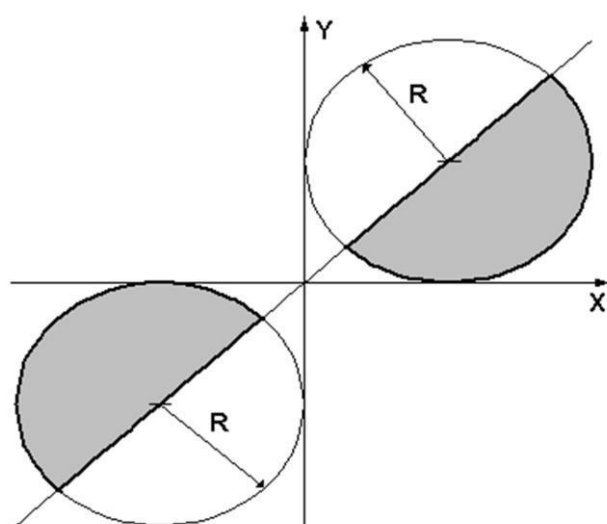
16)



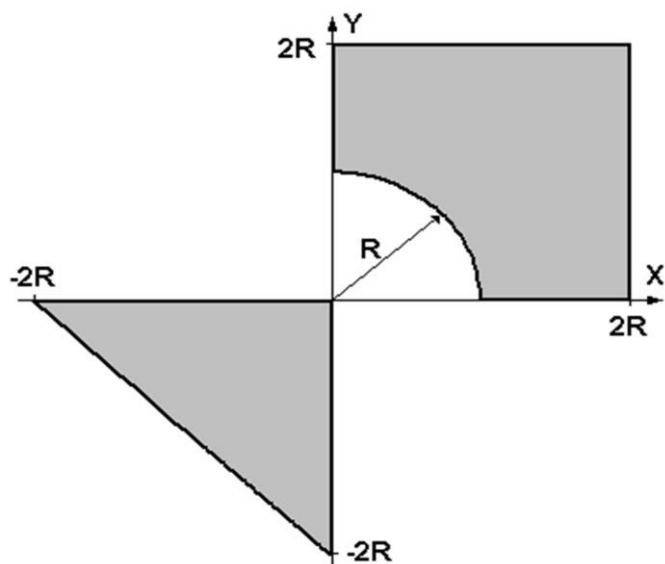
17)



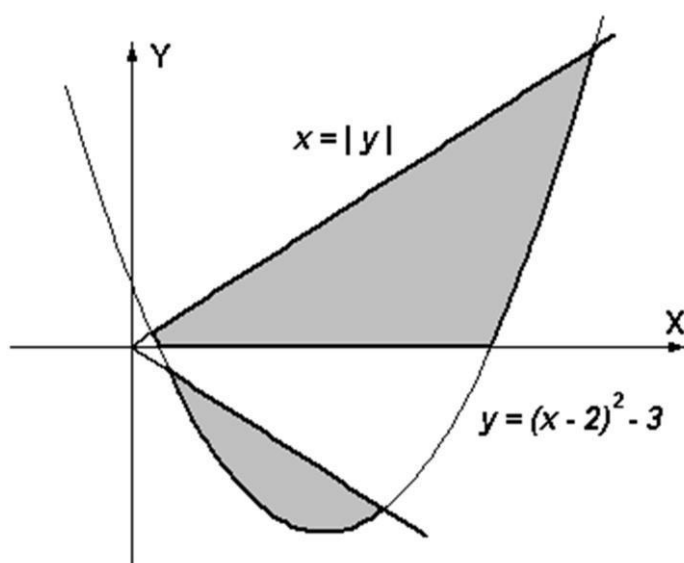
18)



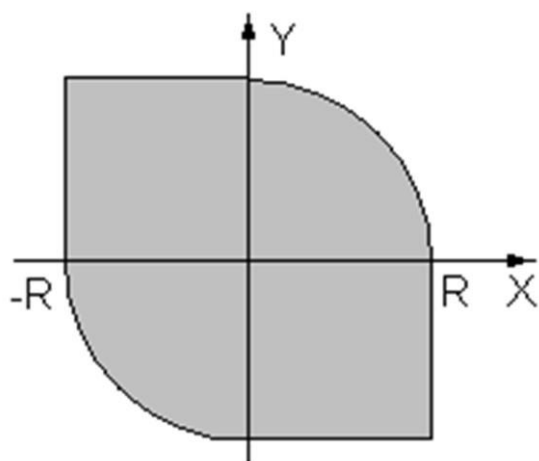
19)



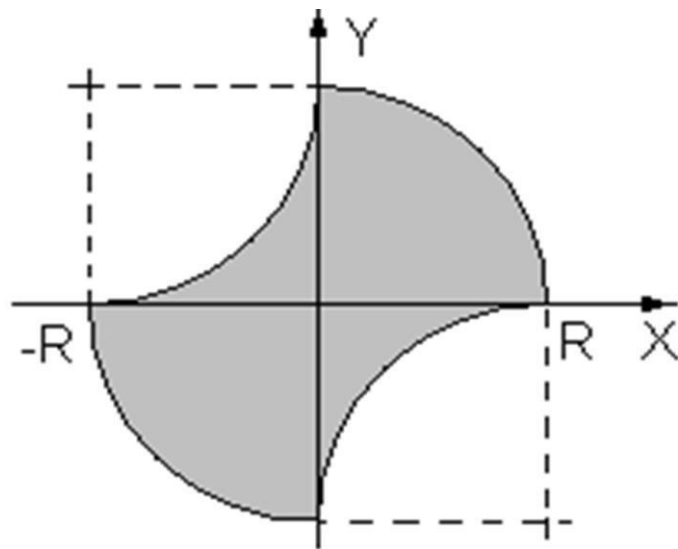
20)



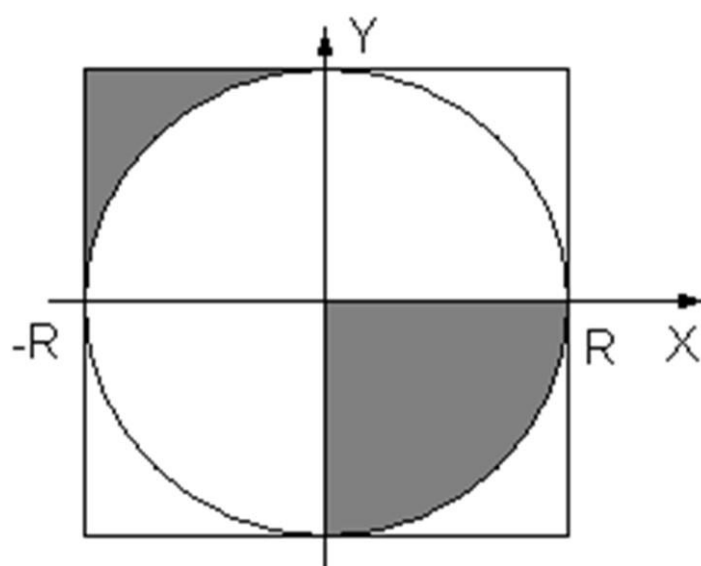
21)



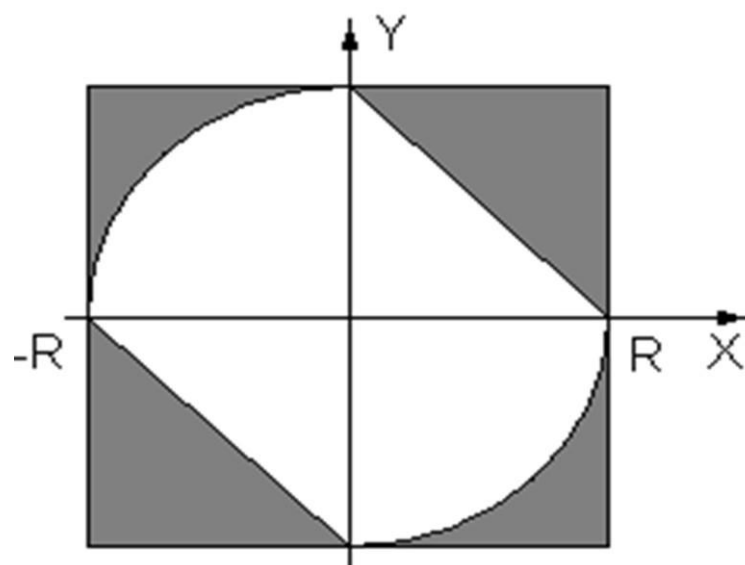
22)



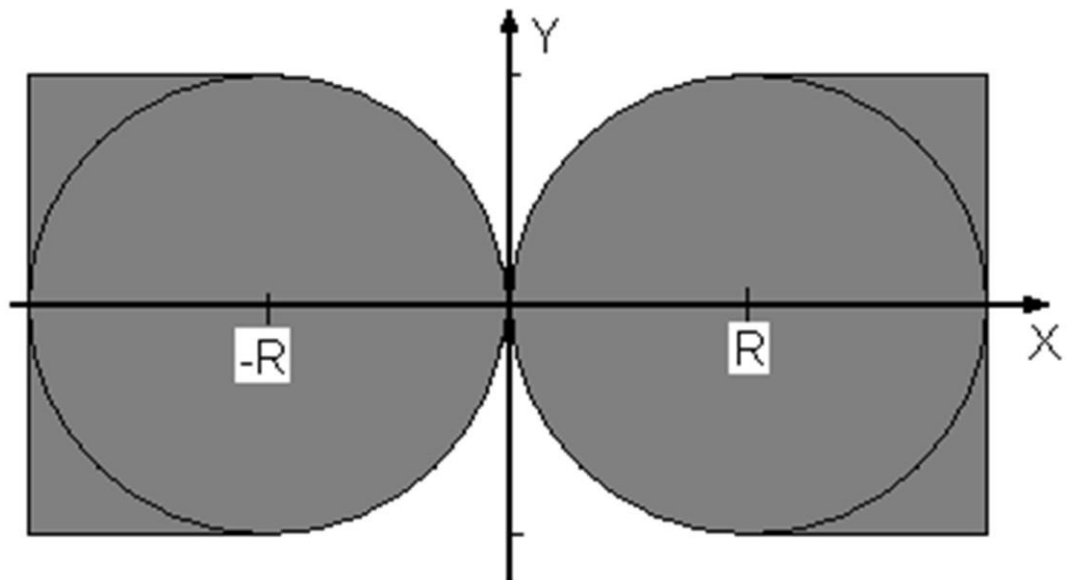
23)



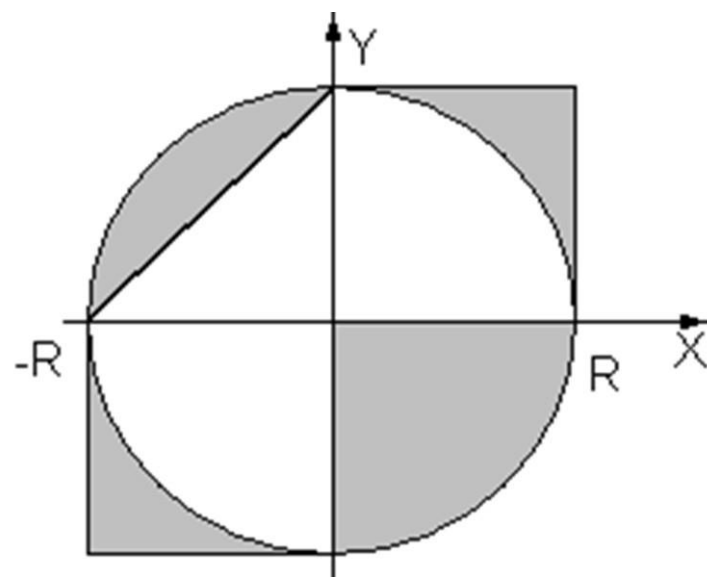
24)



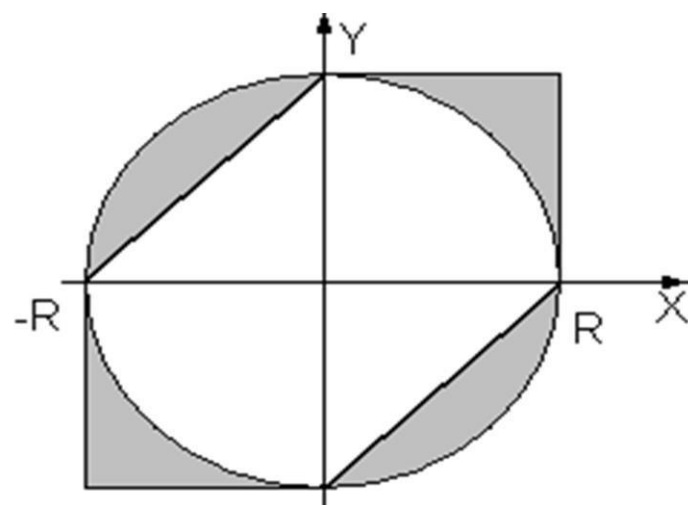
25)



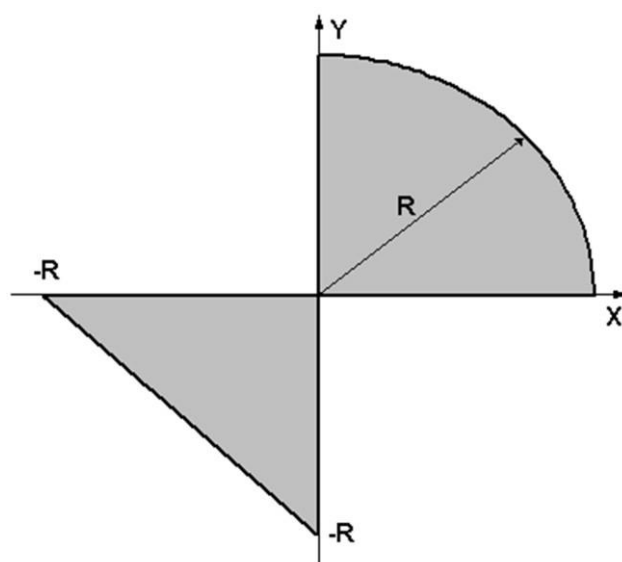
26)



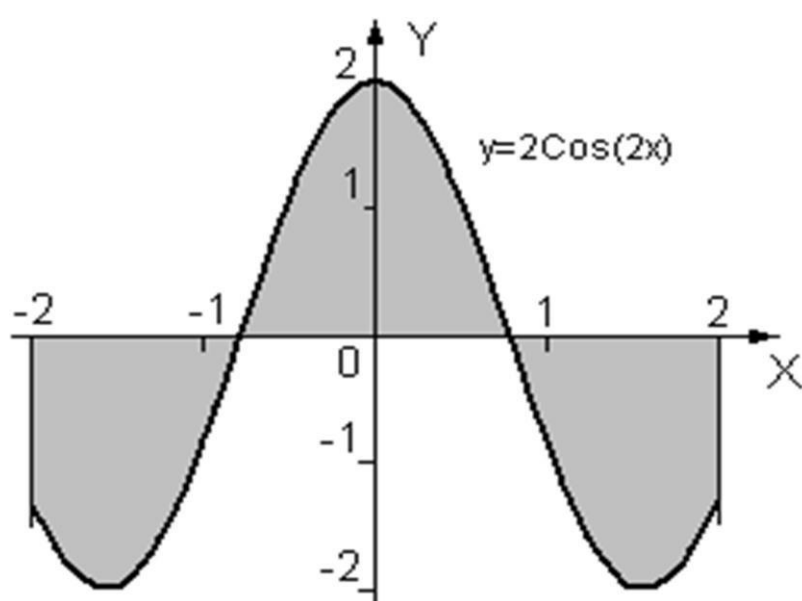
27)



28)



29)



30)

