

# Вариант 13

## Задание 1

```
In[86]:= f[x_] := 36 x3 - 168 x2 + 55 x + 350;
```

```
initialPlot = Plot[f[x], {x, 1, 5}, PlotLabel → "График заданной функции";  
_график функции _пометка графика
```

```
ε = 10-3;
```

```
a = 2;
```

```
b = 3;
```

```
ChordMethod[f_, a_, b_, eps_] := Module[{x0 = a, x1 = b, x2, n = 0},  
_программный модуль
```

```
While[Abs[f[x1]] > eps && Abs[x1 - x0] > eps, x2 = x1 - f[x1] (x1 - x0) / (f[x1] - f[x0]);  
_цикл... _абсолютное значение _абсолютное значение
```

```
  x0 = x1;
```

```
  x1 = x2;
```

```
  n++;];
```

```
{x1, n}];
```

```
{root, iterations} = ChordMethod[f, a, b, ε];
```

```
firstApproximation = Line[{a, f[a]}, {b, f[b]}];  
_(ломаная) линия
```

```
secondApproximationX = b - f[b] (b - a) / (f[b] - f[a]);
```

```
secondApproximation =
```

```
  Line[{a, f[a]}, {secondApproximationX, f[secondApproximationX]}];  
_(ломаная) линия
```

```
Show[initialPlot]
```

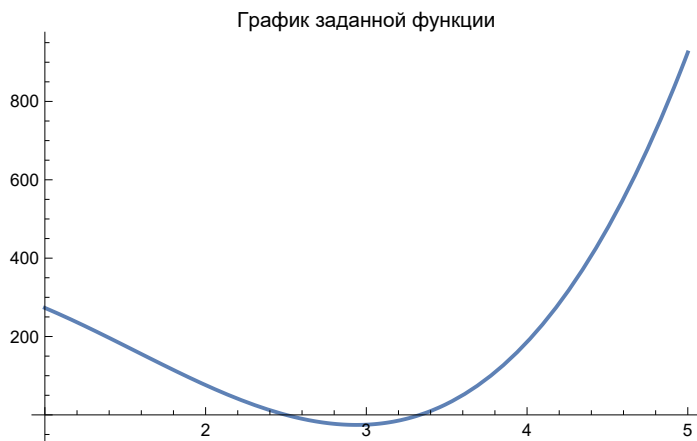
```
_показать
```

```
Show[initialPlot, Graphics[{Red, firstApproximation, Green, secondApproximation}],  
_показать _графика _красный _зелёный
```

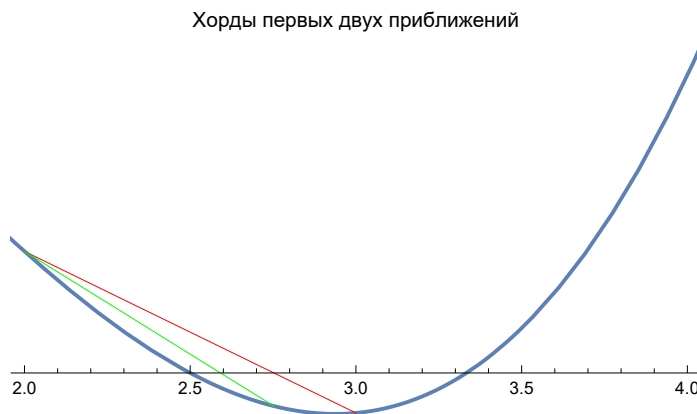
```
PlotRange → {{2, 4}, {-40, 200}}, PlotLabel → "Хорды первых двух приближений"  
_отображаемый диапазон графика _пометка графика
```

```
Print["Найденный корень: ", N[root, 4], "; Число итераций: ", iterations];  
_печатать _численное приближение
```

Out[96]=



Out[97]=



Найденный корень: 2.500; Число итераций: 7

## Задание 2

In[99]:= **Clear[f];**[Очистить](#)

$$f = x^6 + 4x^5 - 10x^4 - 24x^3 + 13x^2 + 44x + 20;$$
**Print["Корни Solve: ", N[Solve[f == 0, x], 1]]**
[печатать](#)[решить ур...](#)[решить уравнения](#)
**Print["Корни NSolve: ", N[NSolve[f == 0, x], 1]]**
[печатать](#)[численное ...](#)[численное решение уравнений](#)
**Print["Корни Roots: ", N[Roots[f == 0, x], 1]]**
[печатать](#)[корни мно...](#)[корни многочлена](#)
**Print["Корень FindRoot: ", N[FindRoot[f == 0, {x, 0}], 1]]**
[печатать](#)[найти корень](#)[найти корень](#)
**Print["Разложение: ", Factor[f]]**
[печатать](#)[факторизовать](#)

Корни Solve: {{x → -5.}, {x → -1.}, {x → -1.}, {x → -1.}, {x → 2.}, {x → 2.}}

Корни NSolve: {{x → -5.}, {x → -1.}, {x → -1.}, {x → -1.}, {x → 2.}, {x → 2.}}

Корни Roots: x == -5. || x == -1. || x == -1. || x == -1. || x == 2. || x == 2.

Корень FindRoot: {x → -0.999997}

 Разложение:  $(-2 + x)^2 (1 + x)^3 (5 + x)$

## Задание 3

In[106]:=

```

Clear[f];
очистить

f[x_] := 15 + 9 x - 4 x2 - 8 (Cos[x + 3])2;
косинус

initialPlot = Plot[f[x], {x, -2, 4}, PlotLabel → "График заданной функции"];
график функции пометка графика

Show[initialPlot]
показать

(* Метод Ньютона *)
ε = 10-3;
maxIterations = 50;
x1 = -2; (* Начальное приближение *)

Do[x2 = x1;
оператор цикла
  x1 = (x1 - f[x1] / f'[x1]) // N;
численное приближение

  If[Abs[x2 - x1] < ε,
... абсолютное значение
    Print["Решение x=", x2 // N, " получено методом Ньютона на ", n, " итерации."];
печатать численное приближение
    Break[];
прервать цикл
  {n, 1, maxIterations}]

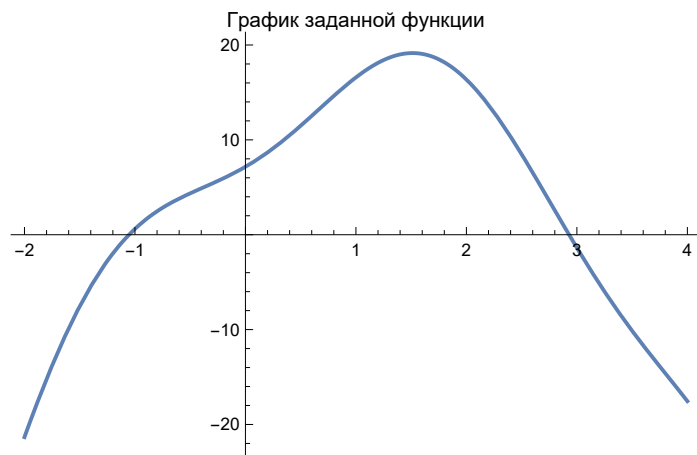
(* Метод Секущих *)
ε = 10-3;
maxIterations = 50;
x1 = -2; (* Начальное приближение *)

Do[x3 = x2; x2 = x1;
оператор цикла
  x1 = (x1 - f[x1] ( (x1 - x3) / (f[x1] - f[x3]) )) // N;
численное приближение

  If[Abs[x2 - x1] < ε,
... абсолютное значение
    Print["Решение x=", x2 // N, " получено методом Секущих на ", n, " итерации."];
печатать численное приближение
    Break[];
прервать цикл
  {n, 1, maxIterations}]

```

Out[109]=



Решение  $x = -1.05367$  получено методом Ньютона на 5 итерации.

Решение  $x = -1.05367$  получено методом Секущих на 2 итерации.

## Задание 4

Уравнение  $15 + 9x - 4x^2 = 8 \cos^2[x + 3]$  нельзя привести к виду  $x = \varphi(x)$ . Следовательно не получится применить метод простых итераций.

## Задание 5

In[118]:=

```
solveRoots = Solve[f[x] == 0, x];
```

└решить уравнения

```
nSolveRoots = NSolve[f[x] == 0, x];
```

└численное решение уравнений

```
findRootRoots = FindRoot[f[x] == 0, {x, 0}];
```

└найти корень

```
Print["Уравнение решается только с помощью функции FindRoot: ", N[findRootRoots, 4]]
```

└печатать

└найти корень

└численное приближение

⋮ Solve: This system cannot be solved with the methods available to Solve. Try Reduce or FindInstance instead.

⋮ NSolve: This system cannot be solved with the methods available to NSolve.

Уравнение решается только с помощью функции FindRoot:  $\{x \rightarrow -1.05366\}$

## Задание 6

In[132]:=

```
Clear[f];
```

```
ОЧИСТИТЬ
```

```
f[x_, y_] =  $\sqrt[3]{(x-1)^2} + \sqrt[3]{(y-3)^2} - 5;$ 
```

```
g[x_, y_] =  $\sqrt{x^2 + y^2} - 2 \text{Cosh}[x - y - 1];$ 
```

```
Гиперболический косинус
```

```
graph1 =
```

```
ContourPlot[f[x, y] == 0, {x, -15, 15}, {y, -15, 15}, Axes → True, Frame → False];
```

```
Контурный график
```

```
Оси
```

```
Истина
```

```
Рамка
```

```
Ложь
```

```
graph2 =
```

```
ContourPlot[g[x, y] == 0, {x, -15, 15}, {y, -15, 15}, Axes → True, Frame → False];
```

```
Контурный график
```

```
Оси
```

```
Истина
```

```
Рамка
```

```
Ложь
```

```
Show[graph1, graph2]
```

```
Показать
```

```
FindRoot[{f[x, y] == 0, g[x, y] == 0}, {x, -1}, {y, -3}]
```

```
Найти корень
```

```
FindRoot[{f[x, y] == 0, g[x, y] == 0}, {x, -2}, {y, -2}]
```

```
Найти корень
```

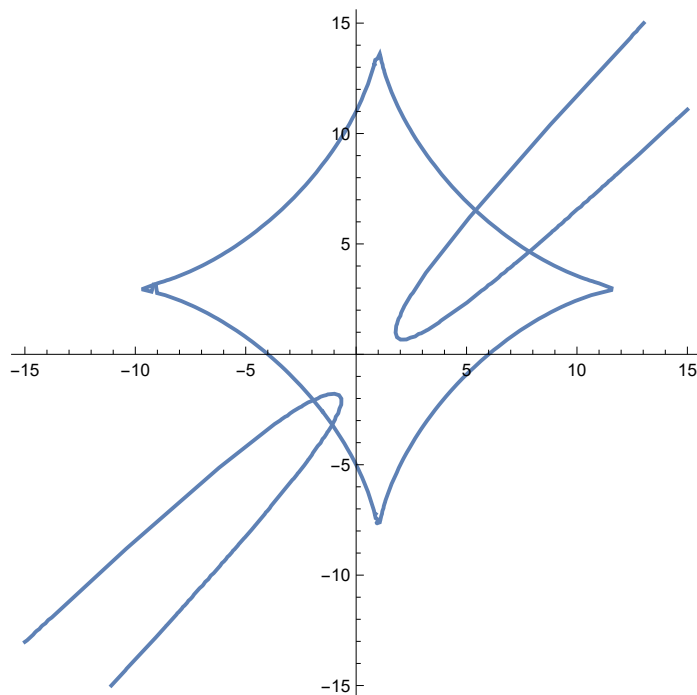
```
FindRoot[{f[x, y] == 0, g[x, y] == 0}, {x, 5}, {y, 6}]
```

```
Найти корень
```

```
FindRoot[{f[x, y] == 0, g[x, y] == 0}, {x, 7}, {y, 4}]
```

```
Найти корень
```

Out[137]:=



Out[138]:=

```
{x → -1.07866, y → -3.18993}
```

Out[139]:=

```
{x → -1.94146, y → -2.05923}
```

Out[140]=

$\{x \rightarrow 5.4, y \rightarrow 6.52196\}$

Out[141]=

$\{x \rightarrow 7.84533, y \rightarrow 4.64728\}$