In[3827]:=

```
(*Лабораторная работа 4
 ст. гр. 221703
 Корнеенко Егор
 Вариант 10 *)

(*Задание 1*)
Clear[x];
```
очистить

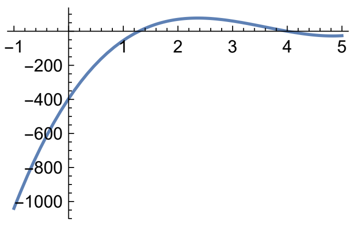
In[3828]:=

$f[x\_] := 14\, x^3 - 151\, x^2 + 479\, x - 396$

In[3829]:=

```
Plot[f[x], {x, -1, 5}, AxesOrigin → {0, 0}, ImageSize → Small]
```
график функции | точка пересечения осей | размер изоб··· | малый

Out[3829]=



In[3830]:=

```
(*Отделяем графически корни алгебраического уравнения*)
```

In[3831]:=

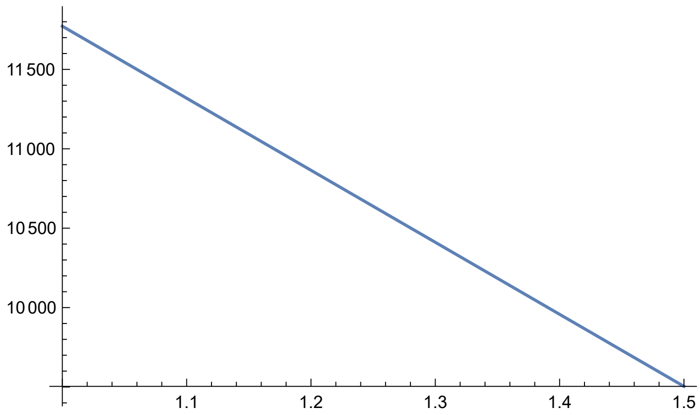
In[3832]:=

```
a = 1; b = 1.5; countOfItters = 1; e = 0.001; maxIters = 100;
(*Находим корень, который находится на данном отрезке*)
```

In[3833]:=

```
Plot[f[a] * f''[x], {x, a, b}](*f[a]*f''[x]>0, следовательно берем первую систему*)
```
график функции

Out[3833]=



In[3834]:=

```
x0 = b;
min = x0;
g[x_] = a - f[a]/(f[x] - f[a]) * (x - a);
sr = g[x0];
```

In[3838]:=

```
While[countOfItters < maxIters,
цикл-пока
  (*Находим корень и колиество итераций с помощью метода хорд*)
     max = g[sr];
  If[(max - sr)^2/Abs[max + min - 2 * sr] < e,
  условный оператор
   Break[];];
   прервать цикл
  countOfItters++;
  min = sr;
  sr = max;
 ];
```

In[3839]:=

```
countOfItters
```

Out[3839]=

```
2
```

In[3840]:=

```
N[max] (*корень*)
численное приближени
```

Out[3840]=

```
1.28656
```

In[3841]:=

```
N[(max - sr)^2/Abs[max + min - 2 * sr]] (*условие окончания итераций выполнилось, меньше e*)
численное приближение
```

Out[3841]=

```
0.000860681
```

In[3842]:=

```
x1 = g[x0]; (*Находим уравнения двух хорд*)
f1 = f[x1];
x2 = g[x1];
f2 = f[x2];
```

In[3846]:=

```
x1
```

Out[3846]=

```
1.32143
```

In[3847]:=

```
f0 = f[x0]
```

Out[3847]=

```
30.
```

In[3848]:=

```
f1
```

Out[3848]=

```
5.5963
```

In[3849]:=

```
f2
```

Out[3849]=

```
0.882788
```

In[3850]:=

$$A = \begin{pmatrix} x1 & 1 \\ x0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} f1 \\ f0 \end{pmatrix}$$

Out[3850]=

```
{{1.32143, 1}, {1.5, 1}}
```

Out[3851]=

```
{{5.5963}, {30.}}
```

In[3852]:=

```
solv = LinearSolve[A, B]
```

решить линейные уравı

Out[3852]=

```
{{136.661}, {-174.991}}
```

In[3853]:=

```
h1[x_] = solv[[1]] * x + solv[[2]] (*1-я хорда*)
```

Out[3853]=

```
{-174.991 + 136.661 x}
```

In[3854]:=

$$A = \begin{pmatrix} x2 & 1 \\ x0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} f2 \\ f0 \end{pmatrix}$$

Out[3854]=

```
{{1.29125, 1}, {1.5, 1}}
```

Out[3855]=

```
{{0.882788}, {30.}}
```

In[3856]:=

```
solv = LinearSolve[A, B]
```

решить линейные уравı

Out[3856]=

```
{{139.481}, {-179.221}}
```

In[3857]:=

```
h2[x_] = solv[[1]] * x + solv[[2]] (*2-я хорда*)
```

Out[3857]=

```
{-179.221 + 139.481 x}
```

In[3858]:=

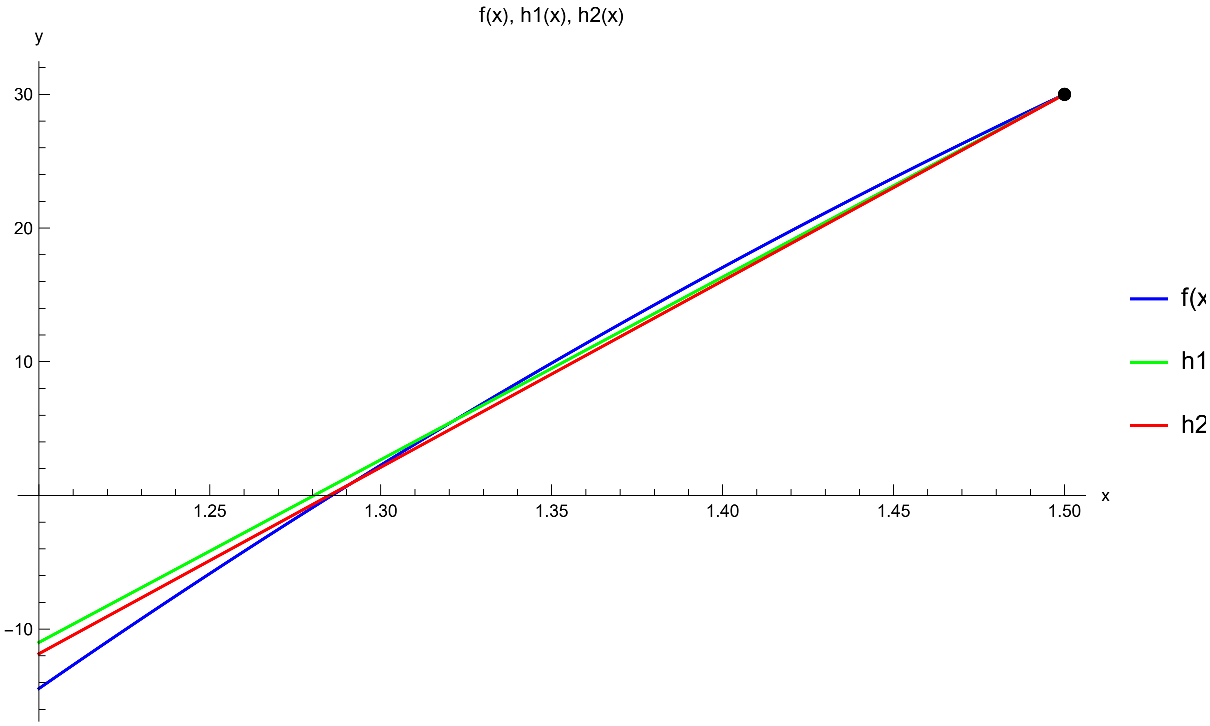
```
(*График функции и хорд*)
```

In[3859]:=

```
Show[Plot[f[x], {x, 1.2, 1.5}, PlotStyle → Blue, PlotLegends → {"f(x)"}],
 Plot[h1[x], {x, 1.2, 1.5}, PlotStyle → Green,
  PlotLegends → {"h1(x) – график первого приближения"}], Plot[h2[x], {x, 1.2, 1.5},
  PlotStyle → Red, PlotLegends → {"h2(x) – график второго приближеня"}],
 AxesLabel → {"x", "y"}, PlotLabel → "f(x), h1(x), h2(x)",
 ImageSize → Large, Epilog → {PointSize[Large], Point[{{x0, f0}}]}]
```

пок⋯ | график функции | стиль графика | синий | легенды графика | график функции | стиль графика | зелёный | легенды графика | график функции | стиль графика | кр⋯ | легенды графика | обозначения на осях | пометка графика | размер изоб⋯ | круп⋯ | эпилог | размер то⋯ | крупный | точка

Out[3859]=



In[3860]:=

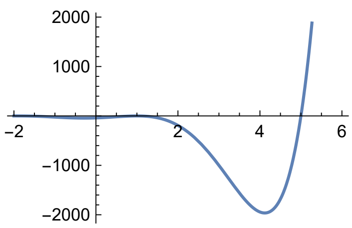
```
(*Задание 2*)
f[x_] := x^6 – x^5 – 19 x^4 – 15 x^3 + 46 x^2 + 28 x – 40
```

In[3861]:=

```
Plot[f[x], {x, -2, 6}, AxesOrigin → {0, 0}, ImageSize → Small]
(*Отделяем графически корни*)
```

график функции | точка пересечения осей | размер изоб⋯ | малый

Out[3861]=



In[3862]:=

```
Solve[f[x] == 0] (*Находим корни с помощью функций пакета Математика*)
```
решить уравнения

Out[3862]=

$\{\{x \to -2\}, \{x \to -2\}, \{x \to -2\}, \{x \to 1\}, \{x \to 1\}, \{x \to 5\}\}$

In[3863]:=

```
NSolve[f[x] == 0]
```
численное решение уравнений

Out[3863]=

$\{\{x \to -2.\}, \{x \to -2.\}, \{x \to -2.\}, \{x \to 1.\}, \{x \to 1.\}, \{x \to 5.\}\}$

In[3864]:=

```
Roots[f[x] == 0, x]
```
корни многочлена

Out[3864]=

$x == -2 \,||\, x == -2 \,||\, x == -2 \,||\, x == 1 \,||\, x == 1 \,||\, x == 5$

In[3865]:=

```
FindRoot[f[x] == 0, {x, -2}]
```
найти корень

Out[3865]=

$\{x \to -2.\}$

In[3866]:=

```
FindRoot[f[x] == 0, {x, 2}]
```
найти корень

Out[3866]=

$\{x \to 1.\}$

In[3867]:=

```
FindRoot[f[x] == 0, {x, 4.5}]
```
найти корень

Out[3867]=

$\{x \to 5.\}$

In[3868]:=

```
Factor[f[x]] (*Раскладываем многочлен f (x) на множители,используя функцию Factor*)
```
факторизовать факторизов

Out[3868]=

$(-5 + x)\,(-1 + x)^2\,(2 + x)^3$

In[3869]:=

```
(*Задание 3*)
```

In[3870]:=

```
f[x_] = Log2[x + 5]
```
двоичный лог

Out[3870]=

$$\frac{\text{Log}[5 + x]}{\text{Log}[2]}$$

In[3871]:=

$g[x\_] = 4 - \sqrt{2 x^2 + 1}$

Out[3871]=

$4 - \sqrt{1 + 2 x^2}$

In[3872]:=

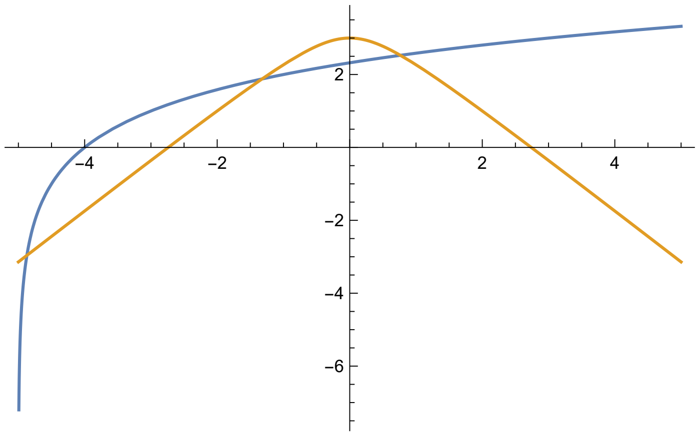
```
Plot[{f[x], g[x]}, {x, -5, 5}]
```

график функции

(*Отделяем графически корни трансцендентного уравнения с помощью функции Plot*)

график функции

Out[3872]=



In[3873]:=

(*Метод Ньютона*)

$p[x\_] = \text{Log2}[x + 5] - 4 + \sqrt{2 x^2 + 1}$

двоичный логарифм

Out[3873]=

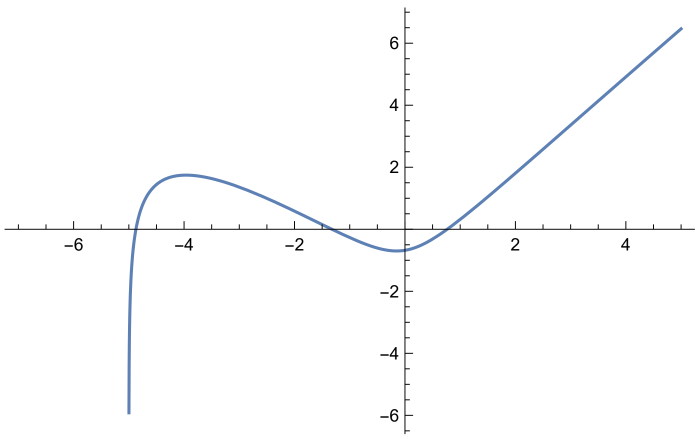
$$-4 + \sqrt{1 + 2 x^2} + \frac{\text{Log}[5 + x]}{\text{Log}[2]}$$

In[3874]:=

```
Plot[p[x], {x, -7, 5}]
```

график функции

Out[3874]=



In[3875]:=

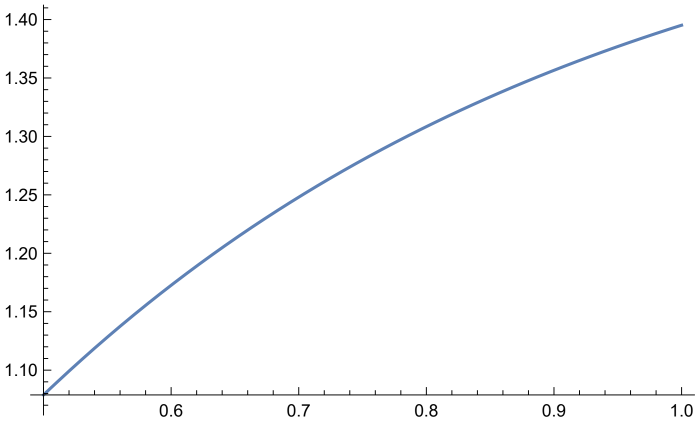
```
a = 0.5; b = 1;(*Проверяем условие f'(x)
и f''(x) сохраняют знак на отрезке[a,b].*)
```

In[3876]:=

```
Plot[p1[x], {x, a, b}]
```

график функции

Out[3876]=



In[3877]:=

```
d[x_] = p''[x]
```

Out[3877]=

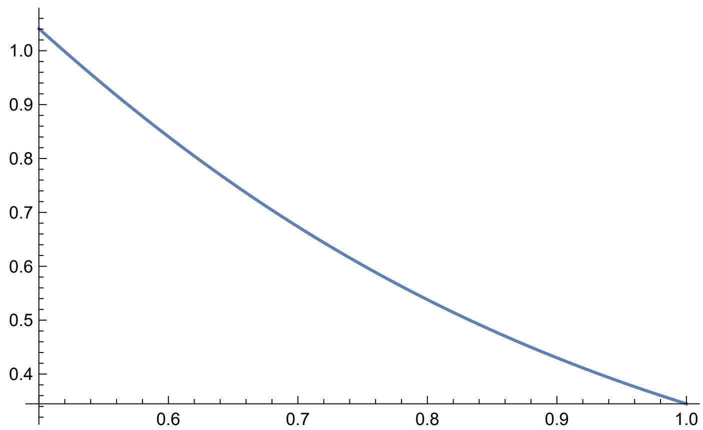
$$-\frac{4 x^2}{\left(1+2 x^2\right)^{3/2}}+\frac{2}{\sqrt{1+2 x^2}}-\frac{1}{(5+x)^2 \text{Log}[2]}$$

In[3878]:=

```
Plot[d[x], {x, a, b}]
```

график функции

Out[3878]=



In[3879]:=

```
N[p[1] * d[1]] (*В качестве начального приближения x следует брать тот
```

численное приближение

```
 конец отрезка [a,b],для которого выполняется условие p[x]*p''[x]>0*)
```

Out[3879]=

0.109314

In[3880]:=

```
countOfItters = 0; e = 0.001; maxIters = 100;
x0 = a (*Находи корень уравнеия и количество итераций с помощью метода Ньютона*)
```

Out[3880]=

0.5

In[3881]:=

```
While[countOfItters < maxIters,
цикл-пока
  countOfItters++;
  x1 = x0 - p[x0]/p'[x0];
  If[Abs[x1 - x0] < e,
  … абсолютное значение
   Break[];];
   прервать цикл
  x0 = x1;
 ];
```

In[3882]:=

```
countOfItters(*количество итераций*)
```

Out[3882]=

```
3
```

In[3883]:=

```
x1(*корень*)
```

Out[3883]=

```
0.764561
```

In[3884]:=

```
FindRoot[p[x] == 0, {x, 0.5}](*Находим корень с помощью функции пакета Математика*)
найти корень
```

Out[3884]=

```
{x → 0.764561}
```

In[3885]:=

```
(*Метод секущих*)
```

In[3886]:=

```
countOfItters = 0; e = 0.001; maxIters = 100;
x0 = a;(*Находим корень уравнения и количество итераций с помощью метода секущих,
начальное приближение берем из метода Ньютона*)
```

In[3887]:=

```
x0 = 0.5;
x1 = 0.6
```

Out[3888]=

```
0.6
```

In[3889]:=

```
While[countOfItters < maxIters,
  countOfItters++;
  x2 = x1 - (x1 - x0)/(p[x1] - p[x0]) * p[x1];
  If[Abs[x1 - x0] < e,
   Break[];];
  x0 = x1;
  x1 = x2;
 ];
```

In[3890]:=

```
countOfItters(*Количество итераций*)
```

Out[3890]=

```
4
```

In[3891]:=

```
x2(*Корень*)
```

Out[3891]=

```
0.764561
```

In[3892]:=

```
(*Задание 4*)
(*1 корень*)
a = 0.5;
b = 1;
```

In[3894]:=

```
1.
```

Out[3894]=

```
1.
```

In[3895]:=

```
M = FindMaximum[p1[x], {x, a, b}]
 (*Находим максимальное значение производной на отрезке [0.5, 1]*)
```

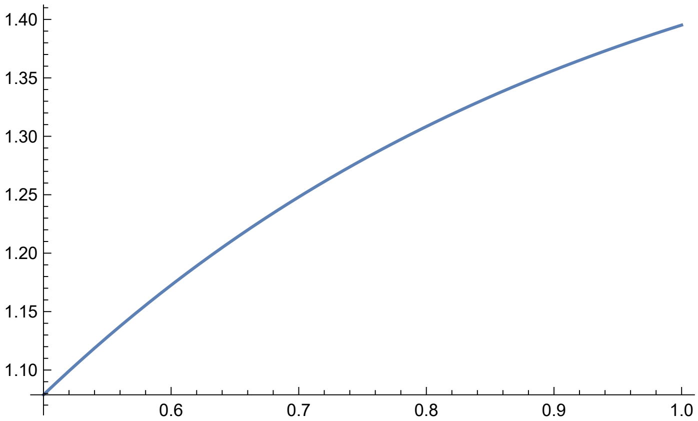
Out[3895]=

```
{1.55691, {x → 3.10053}}
```

In[3896]:=

```
Plot[p1[x], {x, a, b}] (*Смотрим,
 график функции
какой знак у производной на данном отрезке, чтобы определить знак λ*)
```

Out[3896]=



In[3897]:=

```
2 / M[[1]] (*|λ|<2/M – условие для λ*)
```

Out[3897]=

1.2846

In[3898]:=

```
λ = 1 (*Берем λ меньше по модулю этого
  числа и по знаку совпадающее со знаком производной*)
```

Out[3898]=

1

In[3899]:=

```
φ[x_] = x - λ * p[x] (*Переходим к уравнению пригодному для итераций*)
```

Out[3899]=

$$4 + x - \sqrt{1 + 2\,x^2} - \frac{\mathrm{Log}[5 + x]}{\mathrm{Log}[2]}$$

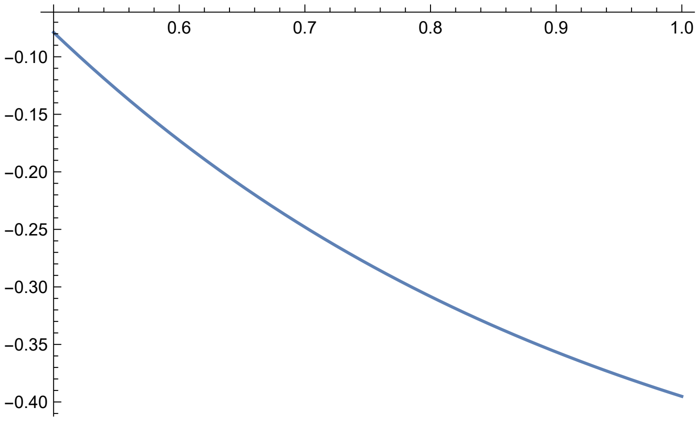
In[3900]:=

```
φ1[x_] = D[φ[x], x]; (*Находим производную и смотрим, чтобы значение
          дифференциировать
 производной на данном отрезке в каждой точке по модулю было меньше единицы*)
```

In[3901]:=

```
Plot[φ1[x], {x, a, b}]
 график функции
```

Out[3901]=



In[3902]:=

```
countOfItters = 0; e = 0.001; maxIters = 100 ; iterations = 0;(*Находим 1-
 ый корень уравнения и количество итераций с помощью метода простой итерации*)
```

In[3903]:=

```
x0 = 0.75;(*x0(начальное приближение) - любое число из отрезка [a, b]*)
```

In[3904]:=

```
For[iterations, iterations < maxIters, iterations++,
цикл ДЛЯ
  x1 = φ[x0];
  If[Abs[x1 - x0] < e,
  ⋯ абсолютное значение
   Break[]];
   прервать цикл
  x0 = x1;];
```

In[3905]:=

```
x1 (*первый корень*)
```

Out[3905]=

0.764462

In[3906]:=

```
iterations(*потребовавшееся число итераций*)
```

Out[3906]=

3

In[3907]:=

$$p[x\_] = \text{Log2}[x+5] - 4 + \sqrt{2x^2+1}$$

двоичный логарифм

Out[3907]=

$$-4 + \sqrt{1+2x^2} + \frac{\text{Log}[5+x]}{\text{Log}[2]}$$

In[3908]:=

```
p1[x_] = p'[x]
```

Out[3908]=

$$\frac{2x}{\sqrt{1+2x^2}} + \frac{1}{(5+x)\,\text{Log}[2]}$$

In[3909]:=

```
a = -2
b = -1
```

Out[3909]=

-2

Out[3910]=

-1

In[3911]:=

```
M = First@FindMaximum[{Abs[p[x]], a ≤ x ≤ b}, x]
    первый найти максимум абсолютное значение
  (*Находим максимальное значение производной на отрезке [-2, -1]*)
```

Out[3911]=

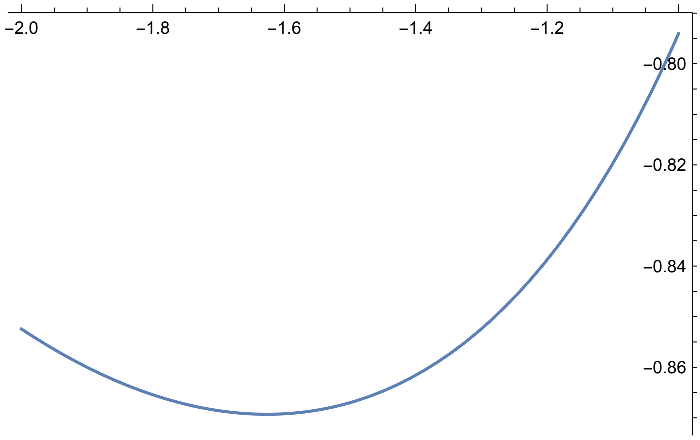
0.267949

In[3912]:=

```
Plot[p1[x], {x, a, b}] (*Смотрим какой знак у производной на данном отрезке,
чтобы определить знак λ*)
```

график функции

Out[3912]=



In[3913]:=

```
2 / M (*|λ|<2/M - условие для λ*)
```

Out[3913]=

7.4641

In[3914]:=

```
λ = -1(*Берем λ меньше по модулю этого
   числа и по знаку совпадающее со знаком производной*)
```

Out[3914]=

−1

In[3915]:=

```
-1
```

Out[3915]=

−1

In[3916]:=

```
-2
```

Out[3916]=

−2

In[3917]:=

```
φ[x_] = x - λ * p[x]
(*Переходим к уравнению пригодному для итераций*)
```

Out[3917]=

$$-4 + x + \sqrt{1 + 2 x^2} + \frac{\text{Log}[5 + x]}{\text{Log}[2]}$$

In[3918]:=

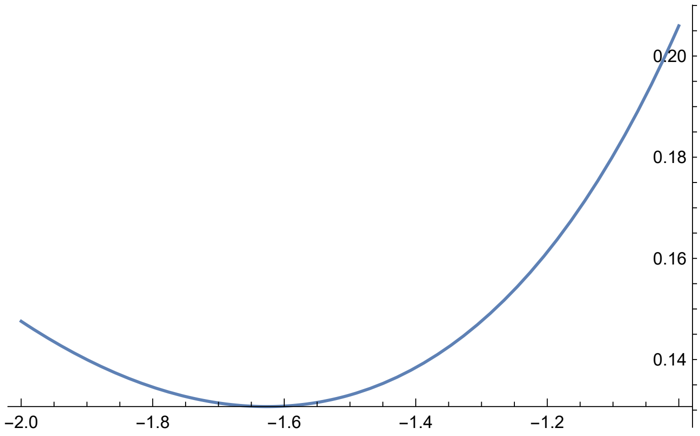
```
φ1[x_] = D[φ[x], x];
(*Находим производную и смотрим, чтобы значение производной
 на данном отрезке в каждой точке по модулю было меньше единицы*)
```

дифференциировать

In[3919]:=

```
Plot[φ1[x], {x, a, b}]
```
график функции

Out[3919]=



In[3920]:=

```
countOfItters = 0; e = 0.001; maxIters = 100 ; iterations = 0;
(*Находим 2-
 ой корень уравнения и количество итераций с помощью метода простой итерации*)
```

In[3921]:=

```
x0 = -1.35;(*x0(начальное приближение) - любое число из отрезка [a, b]*)
```

In[3922]:=

```
For[iterations, iterations < maxIters, iterations++,
  x1 = φ[x0];
  If[Abs[x1 - x0] < e,
   Break[]];
  x0 = x1;];
```
цикл ДЛЯ; абсолютное значение; прервать цикл

In[3923]:=

```
x1(*второй корень*)
```

Out[3923]=

-1.32308

In[3924]:=

```
iterations(*потребовавшееся число итераций*)
```

Out[3924]=

2

In[3925]:=

```
(*Находим 3 корень*)
```

In[3926]:=

```
a = -5
```

Out[3926]=

-5

In[3927]:=

```
b = -4
```

Out[3927]=

-4

In[3928]:=

```
M = First@FindMaximum[{Abs[p[x]], a ≤ x ≤ b}, x]
    первый  найти максимум  абсолютное значение
(*Находим максимальное значение производной на отрезке [-5, -4]*)
```

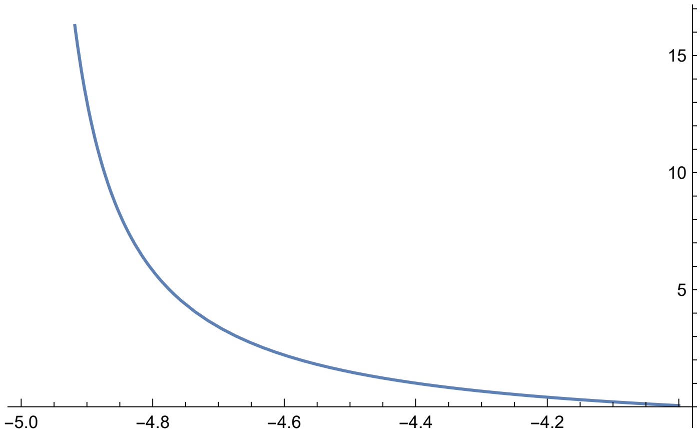
Out[3928]=

```
1.74456
```

In[3929]:=

```
Plot[p1[x], {x, a, b}]
график функции
(*Смотрим, какой знак у производной на данном отрезке, чтобы определить знак λ*)
```

Out[3929]=



In[3930]:=

```
2 / M (*|λ|<2/M - условие для λ*)
```

Out[3930]=

```
1.14642
```

In[3931]:=

```
λ = 0.1
(*Берем λ меньше по модулю этого числа и по знаку совпадающее со знаком производной*)
```

Out[3931]=

```
0.1
```

In[3932]:=

```
φ[x_] = x - λ * p[x]
(*Переходим к уравнению пригодному для итераций*)
```

Out[3932]=

$$x - 0.1\left(-4 + \sqrt{1 + 2 x^2} + \frac{\text{Log}[5 + x]}{\text{Log}[2]}\right)$$

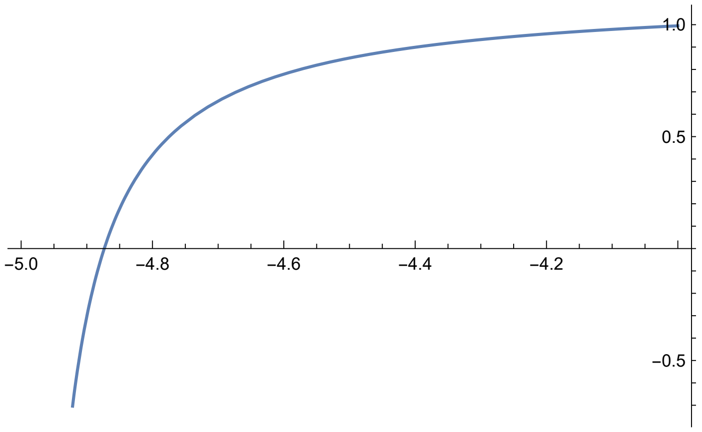
In[3933]:=

```
φ1[x_] = D[φ[x], x];
         дифференцировать
(*Находим производную и смотрим, чтобы значение производной
 на данном отрезке в каждой точке по модулю было меньше единицы*)
```

In[3934]:=

```
Plot[φ1[x], {x, a, b}]
```
график функции

Out[3934]=



In[3935]:=

```
countOfItters = 0; e = 0.001; maxIters = 100 ; iterations = 0;
(*Находим 3-
 ий корень уравнения и количество итераций с помощью метода простой итерации*)
```

In[3936]:=

In[3937]:=

```
x0 = -4.3; (*x0 (начальное приближение) - любое число из отрезка [a, b]*)
```

In[3938]:=

In[3939]:=

```
For[iterations, iterations < maxIters, iterations++,
 x1 = φ[x0];
 If[Abs[x1 - x0] < e,
  Break[];
 x0 = x1;];
```
цикл ДЛЯ; абсолютное значение; прервать цикл

In[3940]:=

In[3941]:=

```
x1 (*третий корень*)
```

Out[3941]=

-4.87163

In[3942]:=

```
iterations (*потребовавшееся число итераций*)
```

Out[3942]=

6

In[3943]:=

```
(*5 задание*)
(*Решаем уравнение из задания 3 с помощью функций пакета Математика*)
Solve[p[x] == 0, Reals]
```
решить уравнения　множество действительных чисел

Out[3943]=

```
{{x → -4.87...}, {x → -1.32...}, {x → 0.765...}}
```

In[3944]:=

```
NSolve[p[x] == 0, Reals]
```
численное решение··· множество дейст

Out[3944]=

```
{{x → -4.87164}, {x → -1.32299}, {x → 0.764561}}
```

In[3945]:=

```
FindRoot[p[x] == 0, {x, -4}]
```
найти корень

Out[3945]=

```
{x → -4.87164}
```

In[3946]:=

```
FindRoot[p[x] == 0, {x, -1}]
```
найти корень

Out[3946]=

```
{x → -1.32299}
```

In[3947]:=

```
FindRoot[p[x] == 0, {x, 1}]
```
найти корень

Out[3947]=

```
{x → 0.764561}
```

In[3948]:=

```
(*Задание 6*)
```

In[3949]:=

```
f[x_, y_] = x^3 + y^3
```

Out[3949]=

$x^3 + y^3$

In[3950]:=

```
g[x_, y_] = 4 - 5 ArcTan[x - 5]
```
арктангенс

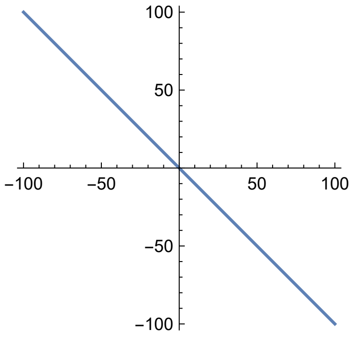
Out[3950]=

```
4 + 5 ArcTan[5 - x]
```

In[3951]:=

```
g1 = ContourPlot[f[x, y] == 0, {x, -100, 100},
     контурный график
  {y, -100, 100}, Axes → True, Frame → False, ImageSize → Small]
                  оси    ист⋯  рамка   ложь   размер изоб⋯ малый
```

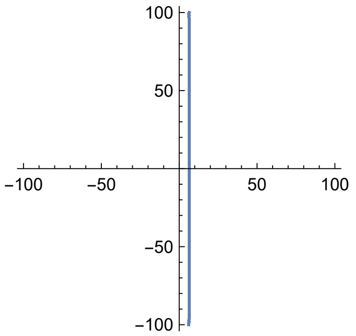
Out[3951]=



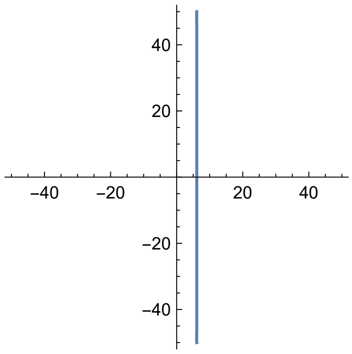
In[3952]:=

```
g2 = ContourPlot[g[x, y] == 0, {x, -100, 100}, {y, -100, 100}, Axes → True,
     контурный график                                          оси    истина
  Frame → False, ImageSize → Small]
  рамка   ложь   размер изоб⋯ малый
```

Out[3952]=



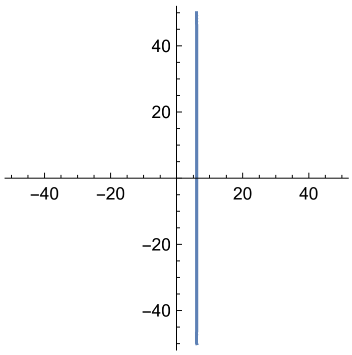
In[3953]:=



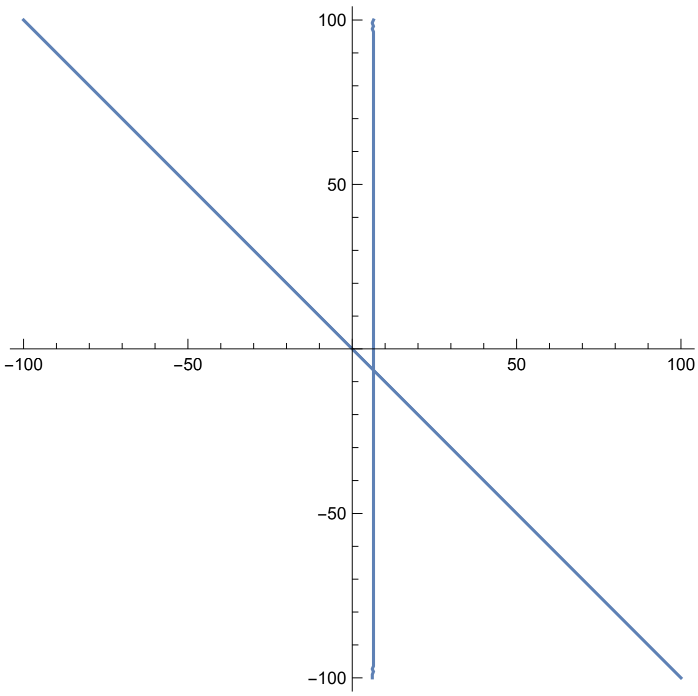

```
Show[g1, g2, ImageSize → Medium]
```

показать размер изоб⋯ средний

Out[3953]=



Out[3954]=



In[3955]:=

```
(*Решаем данную систему с помощью функции пакета Математика*)
```

In[3956]:=

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

найти корень

Out[3956]=

```
{x → 6.02964, y → -6.02964}
```

In[3957]:=

In[3958]:=