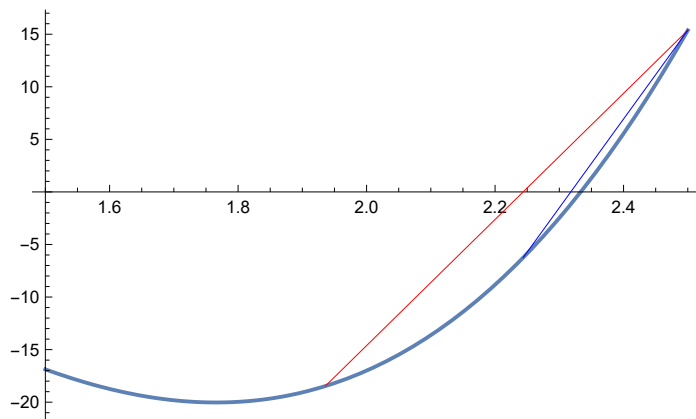


1. Отделите графически корни алгебраического уравнения

$f(x) = 0$ с помощью функции Plot . Найдите один из них (нецелый) с точностью $\xi = 10^{-3}$ методом хорд . Укажите потребовавшееся число итераций . Проиллюстрируйте графически нахождение первых двух приближений (постройте график функции и хорды) .

```
f[x_] := 21 * x^3 - 61 * x^2 + 19 * x + 21
eps = 10^(-3);
a = -1;
b = 2.5;
xPrev = 0;
counter = 0;
Plot[f[x], {x, a, b}, AxesOrigin -> {0, 0}, ImageSize -> Small]
While[ (Abs[a - xPrev] > eps),
  xPrev = a;
  counter++;
  a = a - (f[a] * (b - a) / (f[b] - f[a]));
  If[counter == 2, Print[
    Plot[f[x], {x, 1.5, 2.5}, Epilog -> {Red, Line[{{b, f[b]}, {xPrev, f[xPrev]}]},
    Blue, Line[{{b, f[b]}, {xNext, f[xNext]}]}]}],
  ];
  ];
Print["Root x = ", N[a], ", was acquired on ", counter, " step."]
```



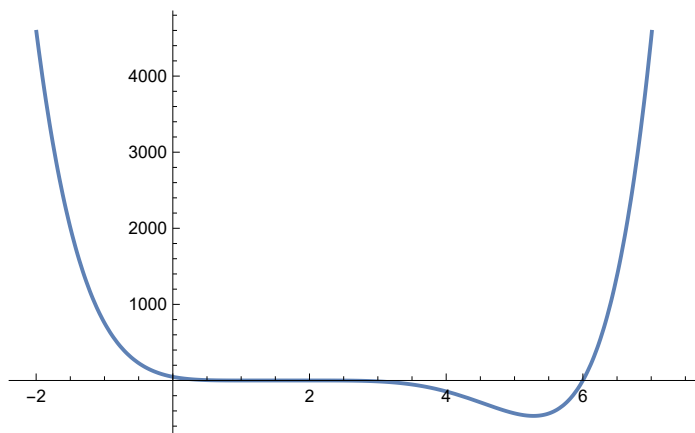
Root x = 2.33327, was acquired on 6 step.

2. Отделите графически и найдите с помощью функций Solve, NSolve, Roots, FindRoot корни алгебраического уравнения $f(x) = 0$. Разложите многочлен $f(x)$ на множители, используя функцию Factor.

```
In[ ]:= f[x_] := x^6 - 14 * x^5 + 73 * x^4 - 188 * x^3 + 256 * x^2 - 176 * x + 48
Plot[f[x], {x, -2.2, 7.5}]
```

график функции

Out[]:=



```
In[ ]:= Solve[f[x] == 0]
```

решить уравнения

Out[]:=

```
{ {x -> 1}, {x -> 1}, {x -> 2}, {x -> 2}, {x -> 2}, {x -> 6} }
```

```
In[ ]:= NSolve[f[x] == 0]
```

численное решение уравнений

Out[]:=

```
{ {x -> 1.}, {x -> 1.}, {x -> 2.}, {x -> 2.}, {x -> 2.}, {x -> 6.} }
```

```
In[ ]:= Roots[f[x] == 0, x]
```

корни многочлена

Out[]:=

```
x == 1 || x == 1 || x == 2 || x == 2 || x == 2 || x == 6
```

```
In[ ]:= FindRoot[f[x] == 0, {x, 6}]
```

найти корень

Out[]:=

```
{x -> 6.}
```

```
In[ ]:= Factor[f[x]]
```

факторизовать

Out[]:=

```
(-6 + x) (-2 + x)^3 (-1 + x)^2
```

Как видно, данное уравнение имеет три корня, причем корни 1 и 2 являются двукратным и трехкратным корнями соответственно

3. Отделите графически корни трансцендентного уравнения с помощью функции Plot. Найдите один из них с точностью $\xi = 10^{-3}$: а) методом Ньютона; б) методом секущих. Укажите потребовавшееся число итераций

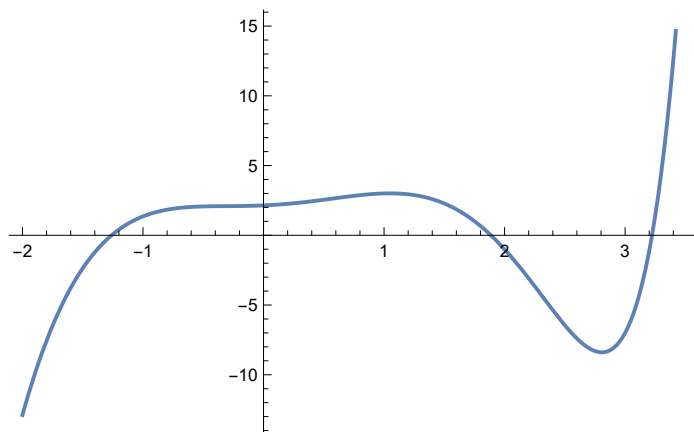
а)

```

In[ ]:= f[x_] := 7^(x-1) + 3^x - x^4 - x + 1
b = -1;
eps = 10^-3;
x1 = b;
Plot[f[x], {x, -2, 3.5}]
[график функции]
Do[x2 = x1; x1 = (x1 - f[x1] / f'[x1]) // N;
[оператор цикла] [численное приближение]
If[Abs[x2 - x1] < eps, Print["Решение x=", x2 // N, " получено на ", n " шаге."];
[... [абсолютное значение] [печатать] [численное приближение]
Break[]], {n, 1, 100}]
[прервать цикл]

```

Out[]=



Решение $x = -1.26082$ получено на 4 шаге.

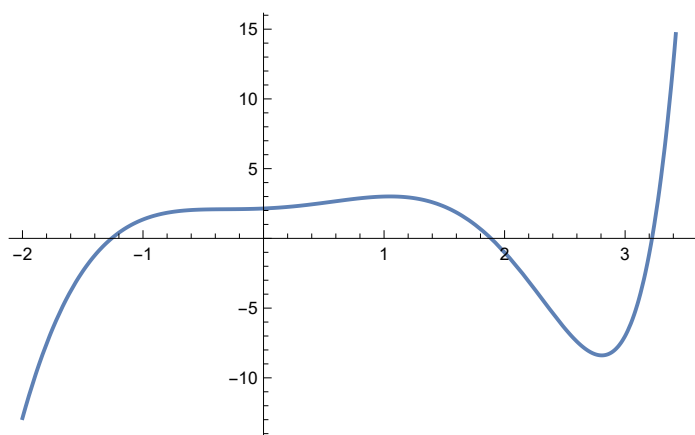
b)

```

In[ ]:= f[x_] := 7^(x-1) + 3^x - x^4 - x + 1
b = -1;
eps = 10^-3;
x1 = b;
x3 = x1;
x1 = (x1 - f[x1] / f'[x1]);
Plot[f[x], {x, -2, 3.5}]
[график функции]
Do[x2 = x1; x1 = x1 - ((x1 - x3) * f[x1] / (f[x1] - f[x3])) // N; x3 = x2;
[оператор цикла] [численное приближение]
If[Abs[x2 - x1] < eps, Print["Решение x=", x2 // N, "получено на ", n " шаге."];
[... [абсолютное значение] [печатать] [численное приближение]
Break[]], {n, 1, 100}]
[прервать цикл]

```

Out[]:=



Решение $x = -1.26085$ получено на 4 шаге.

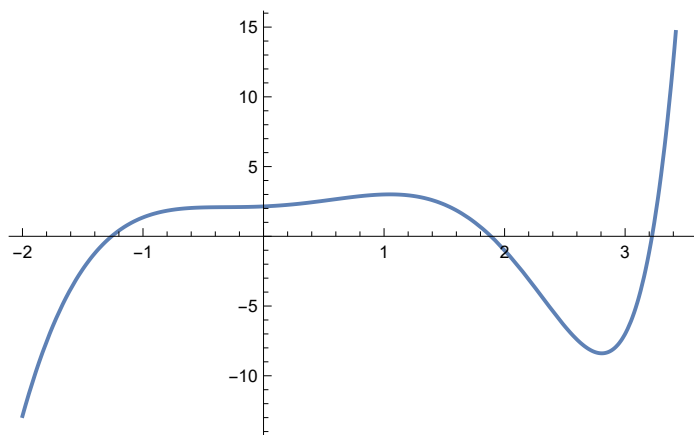
4. Приведите уравнение (3.1 – 3.16) к виду, пригодному для итераций. Найдите его корни методом простых итераций с точностью $\xi = 10^{-3}$.
Укажите потребовавшееся число итераций.

```

f[x_] := 7^(x-1) + 3^x - x^4 - x + 1
a = -2;
b = -1;
eps = 10^-3;
x1 = b;
f2[x_] = f'[x];
Plot[f[x], {x, -2, 3.5}]
[график функции]
Do[x2 = x1; λ = 1 / f2[x1]; x1 = x1 - λ * f[x1];
[оператор цикла]
If[Abs[x2 - x1] < eps, Print["Решение x=", x2 // N, "получено на ", n " шаге."];
[... [абсолютное значение] [печатать] [численное приближение]
Break[]], {n, 1, 100}]
[прекратить цикл]

```

Out[8]=



Решение $x = -1.26082$ получено на 4 шаге.

5. Решите уравнение (3.1 – 3.16) с помощью функций Solve, NSolve, FindRoot.

```
f[x_] := 7^(x-1) + 3^x - x^4 - x + 1
```

```
In[8]:= Solve[f[x] == 0 && -10 < x < 10, x]
[решить уравнения]
```

Out[8]=

```
{ {x -> -1.26...}, {x -> 1.89...}, {x -> 3.22...} }
```

```
In[9]:= NSolve[f[x] == 0 && -10 < x < 10, x]
[численное решение уравнений]
```

Out[9]=

```
{ {x -> -1.26032}, {x -> 1.88718}, {x -> 3.22319} }
```

```
In[10]:= FindRoot[f[x] == 0, {x, 4}]
[найти корень]
```

Out[10]=

```
{x -> 3.22319}
```

6. Дана система двух нелинейных уравнений $f(x, y) = 0$, $g(x, y) = 0$. Используя средства пакета Mathematica, изобразите на одном чертеже кривые $f(x, y) = 0$ и $g(x, y) = 0$, и решите данную

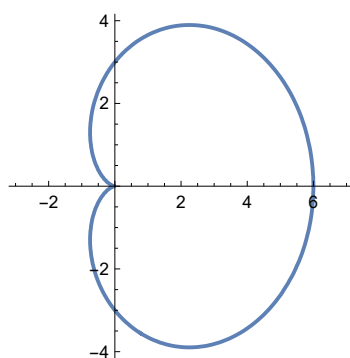
систему.

```

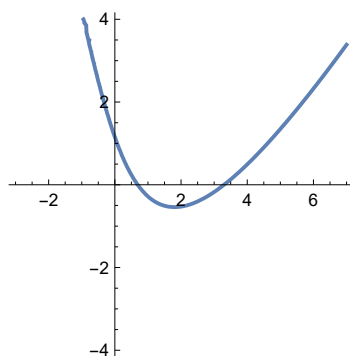
In[ ]:= f[x_, y_] := (x^2 + y^2 - 3 * x) ^2 - 9 * (x^2 + y^2)
g[x_, y_] := 2 * y - 3 * x + 5 * Log[4 * x + y] - 3
                                         [натуральный логарифм]
gr1 = ContourPlot[f[x, y] == 0, {x, -3, 7},
                  [контурный график]
                  {y, -4, 4}, Axes → True, Frame → False, ImageSize → Small]
                                         [оси] [истина] [рамка] [ложь] [размер изобр... [малый]
gr2 = ContourPlot[g[x, y] == 0, {x, -3, 7},
                  [контурный график]
                  {y, -4, 4}, Axes → True, Frame → False, ImageSize → Small]
                                         [оси] [истина] [рамка] [ложь] [размер изобр... [малый]

```

Out[]=



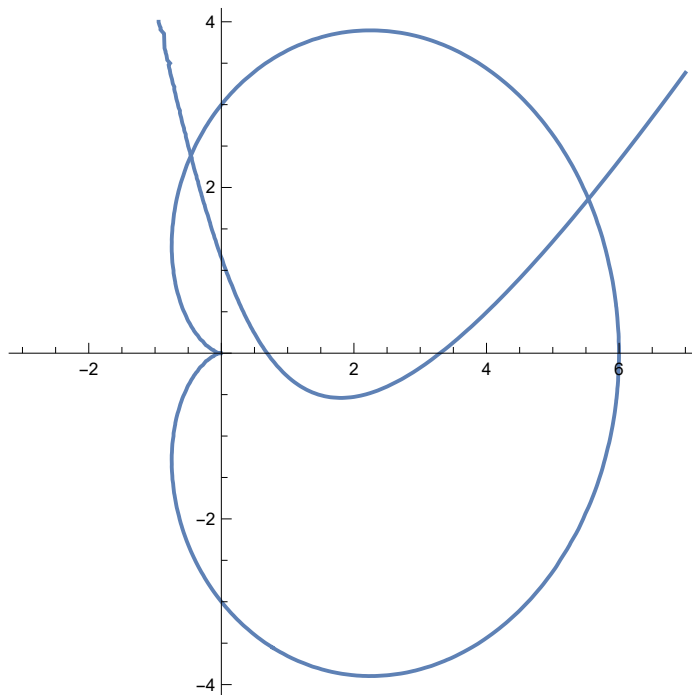
Out[]=



```
In[ ]:= Show[gr1, gr2, ImageSize -> Medium]
```

[показать](#) [размер изобра...](#) [средний](#)

```
Out[ ]:=
```



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

[найти корень](#)

```
Out[ ]:=
```

```
{x -> -0.462892, y -> 2.38354}
```

```
In[ ]:= FindRoot[{f[x, y] == 0, g[x, y] == 0}, {x, 6}, {y, 2}]
```

[найти корень](#)

```
Out[ ]:=
```

```
{x -> 5.53919, y -> 1.86172}
```