

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Математический факультет

Кафедра теории функций и геометрии

Программная реализация (на языке JavaScript) алгоритмов
генерации ФОС по математике 2023

Курсовая работа

Направление 010501 Фундаментальные математика и механика

Зав.кафедрой _____ д.физ.-мат.н., проф. Е.М. Семёнов

Обучающийся _____ А.С. Суматохина

Руководитель _____ д.физ.-мат.н., проф. Е.М. Семёнов

Воронеж 2023

Содержание

Введение	3
1 Глава первая	4
1.1 Реализация алгоритмов на базе проекта Час ЕГЭ	4
1.2 Преимущества программной генерации заданий	13
2 Глава вторая	14
2.1 Функции, используемые в проекте	14
2.2 Работа со строками	16
2.3 Вклад автора в расширение каталога	17
Заключение	40

Введение

Единый государственный экзамен (ЕГЭ) — централизованно проводимый в Российской Федерации экзамен в средних учебных заведениях — школах, лицеях и гимназиях, форма проведения ГИА (Государственный Итоговая Аттестация) по образовательным программам среднего общего образования. Служит одновременно выпускным экзаменом из школы и вступительным экзаменом в вузы.

За два года подготовки к ЕГЭ школьники сталкиваются с дефицитом заданий для подготовки. А учителя со списыванием ответов при решении задач экзамена учениками. Также в конце 2021 года в список заданий ЕГЭ были добавлены новые задания под номером 9, количество которых для прорешивания очень мало. Помимо перечисленного существуют задания, на решение которых требуется менее минуты, а их составление вручную занимает несоразмерно много времени. Проект «Час ЕГЭ» позволяет решить все эти проблемы.

«Час ЕГЭ» — компьютерный образовательный проект, разрабатываемый при математическом факультете ВГУ в рамках «OpenSource кластера» и предназначенный для помощи учащимся старших классов подготовиться к тестовой части единого государственного экзамена. Задания в «Час ЕГЭ» генерируются случайным образом по специализированным алгоритмам, называемых шаблонами, каждый из которых охватывает множество вариантов соответствующей ему задачи. Для пользователей предназначены четыре оболочки (режима работы): «Случайное задание», «Тесты на печать», «Полный тест» и «Мини-интеграция». «Час ЕГЭ» является полностью открытым (код находится под лицензией GNU GPL 3.0) и бесплатным. В настоящее время в проекте полностью реализованы тесты по математике с кратким ответом (бывшая «часть В»). [4] Планируется с течением времени включить в проект тесты по другим предметам школьной программы.

«Мини-интеграция» — это форма сотрудничества с образовательными интернет-ресурсами, при которой учебно-методический материал на странице ресурса дополняется виджетами тренажера с заданиями, соответствующими теме статьи, для возможности практического применения полученных знаний. В настоящее время достигнуто сотрудничество с двумя образовательными ресурсами: ege-ok.ru и matematikalegko.ru.

1. Глава первая

1.1. Реализация алгоритмов на базе проекта Час ЕГЭ

Как было уже сказано ранее, генерация заданий происходит при помощи шаблонов. Программная реализация на языке JavaScript обеспечивает генерацию текстового задания, графической на основе canvas ¹ в случае необходимости, ответа и иногда подробного решения.

Алгоритм написания простого шаблона [3] на основе номера 109083 Открытого Банка Заданий ЕГЭ [4]:

1. Выбираем задание из Открытого Банка Заданий ЕГЭ и копируем его текст.
2. Добавляем ответ в поле answers (по умолчанию 0).

```
1 (function () {  
2     NAinfo.requireApiVersion(0, 2);  
3     NATask.setTask({  
4         text: 'Изюм получается в процессе сушки винограда. Сколько килограммов винограда  
        потребуется для получения 2 килограммов изюма, если виноград содержит 90% воды, а изюм  
        содержит 5% воды?',  
5         answers: 0,  
6     });  
7 })();
```

3. Инициализируем всех необходимые переменные для задачи (вес, проценты и так далее). Присваиваем им значения при помощи функции sluchch() или slKrome() (см. главу 2). Для хранения ответа создаём отдельную переменную.

```
1 (function () {  
2     NAinfo.requireApiVersion(0, 2);  
3     let ves = sluchch(10, 100);  
4     let pr1, pr2, answ;  
5     do {  
6         pr1 = sluchch(4, 20);  
7         pr2 = sluchch(70, 90);  
8         answ = (ves * (100 - pr1)) / (100 - pr2);  
9     } while (!(answ * 100).isZ());  
10    NATask.setTask({  
11        text: 'Изюм получается в процессе сушки винограда. Сколько килограммов винограда  
        потребуется для получения 2 килограммов изюмаесли, виноград содержит 90 % воды, изюм  
        содержит 5 % воды ? ',  
12        answers: 0,  
13    });  
14 })();
```

4. Заменяем все числа в тексте на переменные (при помощи +'.'+).

¹HTML элемента, использующегося для рисования графики средствами языков программирования и алгоритма решения задачи

5. Обособляем слова, которые не влияют на условия задачи. Это могут быть имена, профессии, транспорт и т.п.
6. Создаём переменные, которые будут отвечать за выбранные в прошлом пункте слова, и заменяем слова на переменные в тексте задачи. Выбираем их значения из массивов при помощи `iz()` (см. главу 2).
7. Иногда в задании выбранные слова используются в разных падежах. Для этого в проекте существует лексический модуль. Используем на склоняемых словах функцию `sklonlxxand()`. Теперь необходимо указать падеж слов в задании. Также при необходимости заглавной буквы в слове используем `toZagl()` (см. главу 2).
8. Если в тексте задачи присутствуют слова, зависящие от числительных, к ним применяется функция `chislitlx()` (см. главу 2).

```
1 (function () {  
2     'use strict';  
3     NAinfo.requireApiVersion(0, 2);  
4     let ves = sluchch(10, 100);  
5     let pr1, pr2, answ;  
6     do {  
7         pr1 = sluchch(4, 20);  
8         pr2 = sluchch(70, 90);  
9         answ = (ves * (100 - pr1)) / (100 - pr2);  
10    } while (!(answ * 100).isZ());  
11    let pair = sklonlxxand([  
12        ['абрикос', 'курага'],  
13        ['виноград', 'изюм'],  
14        [  
15            ['финик', 'яблоко', 'груша', 'банан', 'дыня', 'персик',  
16                'инжир', 'манго', 'папайя', 'хурма', 'ананас', 'кокос',  
17                'шиповник', 'клюква', 'барбарис', 'клубника', 'вишня'  
18            ].iz(),  
19            'сухофрукт'  
20        ],  
21    ].iz());  
22    NATask.setTask({  
23        text: 'При сушке ' + pair[0].re + ' получается ' +  
24            pair[1].ie + '. Сколько килограммов ' + pair[0].re +  
25            ' потребуется для получения ' +  
26            ves + ' килограммов ' + pair[1].re + ', если ' +  
27            pair[0].ie + ' содержит ' + pr2 + '% воды, а ' +  
28            pair[1].ie + ' содержит ' + pr1 + '% воды ? ',  
29        answers: answ,  
30    });  
31 })();
```

Но не все задачи решаются при помощи линейного алгоритма. Тогда в коде шаблона используется слишком много циклов с постусловиями (или предусловиями). Для таких случаев в проекте существует функция `retryWhileUndefined()`, которое будет запускать программу до тех пор, пока случайные переменные не будут удовлетворять всем условиям. Перезапуск осуществляется посредством постановки условий с последующим `return`. Рассмотрим такой пример.

Алгоритм написания шаблона с изображением графика прямой:

1. Для отображения рисунка необходимо вызвать функцию `chas2.task.modifiers.addCanvasIllustration()`, которой передаются высота и ширина изображения и функция отрисовки.

```
1 retryWhileUndefined(function () {
2     NAinfo.requireApiVersion(0, 2);
3     chas2.task.modifiers.addCanvasIllustration({
4         width: 300,
5         height: 300,
6         paint: paint1,
7     });
8     return true;
9 }, 100000);
```

2. Пусть наша прямая проходит через две точки (x_1, y_1) и (x_2, y_2) . Инициализируем их координаты случайными целыми числами. Теперь через них выразим коэффициенты уравнения прямой $y = kx + b$. Для дальнейшего вычисления точек прямой напомним вспомогательную функцию $f(x)$.

```
1 retryWhileUndefined(function () {
2     NAinfo.requireApiVersion(0, 2);
3
4     function f(x) {
5         return k * x + b;
6     }
7     let x1 = sluchch(-6, 6);
8     let y1 = sluchch(-7, 6);
9     let x2 = sluchch(-10, 10);
10    let y2 = sluchch(8, 20).pm();
11
12    let k = (y1 - y2) / (x1 - x2);
13    if (!k.isZ() && k != 0)
14        return;
15    let b = y1 - k * x1;
16    chas2.task.modifiers.addCanvasIllustration({
17        width: 300,
18        height: 300,
19        paint: paint1,
20    });
21    return true;
22 }, 100000);
```

3. Для того, чтобы уравнения нашей прямой можно было восстановить по рисунку, необходимо и достаточно, чтобы были видны две её точки. Для поиска таких точек используем функцию `intPoints()` (см. главу 2).

```
1 retryWhileUndefined(function () {
2     NAinfo.requireApiVersion(0, 2);
3
4     function f(x) {
5         return k * x + b;
6     }
7     let x1 = sluchch(-6, 6);
8     let y1 = sluchch(-7, 6);
```

```

9      let x2 = sluchch(-10, 10);
10     let y2 = sluchch(8, 20).pm();
11     retryWhileUndefined(function () {
12         NAinfo.requireApiVersion(0, 2);
13
14         function f(x) {
15             return k * x + b;
16         }
17         let x1 = sluchch(-6, 6);
18         let y1 = sluchch(-7, 6);
19         let x2 = sluchch(-10, 10);
20         let y2 = sluchch(8, 20).pm();
21
22         let k = (y1 - y2) / (x1 - x2);
23         if (!k.isZ() && k != 0)
24             return;
25         let b = y1 - k * x1;
26         let points = intPoints(f, {
27             minX: -5,
28             maxX: 5,
29             minY: -5.5,
30             maxY: 5.5,
31             step: 1,
32         });
33         if (points.length < 2)
34             return;
35         chas2.task.modifiers.addCanvasIllustration({
36             width: 300,
37             height: 300,
38             paint: paint1,
39         });
40         return true;
41     }, 100000);
42
43     let k = (y1 - y2) / (x1 - x2);
44     if (!k.isZ() && k != 0)
45         return;
46     let b = y1 - k * x1;
47     chas2.task.modifiers.addCanvasIllustration({
48         width: 300,
49         height: 300,
50         paint: paint1,
51     });
52     return true;
53 }, 100000);

```

4. Приступим к изображению графика. Инициализируем переменную `paint1` как функцию `function(ct)`. Внутри неё зададим высоту и ширину графика в пикселях. Изображаем координатную плоскость функцией `drawCoordPlane()` и сам график функцией `graph9AdrawFunction()`. А также отметим две целые точки посредством `graph9AmarkCircles()` (см. главу 2).

```

1  retryWhileUndefined(function () {
2      NAinfo.requireApiVersion(0, 2);
3
4      function f(x) {
5          return k * x + b;

```

```

6      }
7      let x1 = sluchch(-6, 6);
8      let y1 = sluchch(-7, 6);
9      let x2 = sluchch(-10, 10);
10     let y2 = sluchch(8, 20).pm();
11
12     let k = (y1 - y2) / (x1 - x2);
13     if (!k.isZ() && k != 0)
14         return;
15     let b = y1 - k * x1;
16
17     let points = intPoints(f, {
18         minX: -5,
19         maxX: 5,
20         minY: -5.5,
21         maxY: 5.5,
22         step: 1,
23     });
24     if (points.length < 2)
25         return;
26     let paint1 = function (ct) {
27         let h = 300;
28         let w = 300;
29         //Оси координат
30         ct.drawCoordPlane(w, h, {
31             hor: 1,
32             ver: 1
33         }, {
34             x1: '1',
35             y1: '1',
36             sh1: 13,
37         }, 20);
38         //График
39         ct.scale(20, -20);
40         ct.lineWidth = 0.1;
41         graph9AdrawFunction(ct, f, {
42             minX: -6.5,
43             maxX: 7,
44             minY: -7,
45             maxY: 6,
46             step: 0.05,
47         });
48         //точки
49         graph9AmarkCircles(ct, points, 2, 0.05);
50
51     };
52     chas2.task.modifiers.addCanvasIllustration({
53         width: 300,
54         height: 300,
55         paint: paint1,
56     });
57     return true;
58 }, 100000);

```

Алгоритм написания шаблона с изображением графика производной:

1. Для отображения рисунка необходимо вызвать функцию `chas2.task.modifiers.addCanvasIllustration()`, которой передаются высота и ширина изображения

и функция отрисовки.

```
1 (function() {
2   retryWhileError(function() {
3     'use strict';
4     chas2.task.modifiers.addCanvasIllustration({
5       width: 500,
6       height: 400,
7       paint: paint1,
8     });
9   });
10 })();
```

2. Определяем область определения будущей функции и заполняем два массива, где будут находиться координаты точек, через которые будет проходить график функции. Строим график по точкам при помощи библиотеки `cubic-spline`.

```
1 (function () {
2   retryWhileError(function () {
3     'use strict';
4
5     function f(x) {
6       return spline.at(x);
7     }
8     NAinfo.requireApiVersion(0, 2);
9     let maxX = sl(8, 10);
10    let minX = maxX - sl(13, 18);
11    let X = [];
12    let Y = [];
13    for (let i = minX; i <= maxX; i += sl(1, 2))
14      X.push(i);
15    Y.push(sl(1, 6).pm());
16    for (let i = 1; i < X.length; i++) {
17      do {
18        if (i % 2)
19          Y[i] = Y[i - 1] + sl(1, 4).pm();
20        else
21          Y[i] = 0;
22      } while (Y[i].abs() > 5);
23    }
24    let spline = new Spline(X, Y);
25
26    chas2.task.modifiers.addCanvasIllustration({
27      width: 500,
28      height: 400,
29      paint: paint1,
30    });
31  });
32 })();
```

3. Находим экстремумы функции, промежутки её возрастания и убывания.

```
1 (function() {
2   retryWhileError(function() {
```

```

3      'use strict';
4
5      function f(x) {
6          return spline.at(x);
7      }
8      NAinfo.requireApiVersion(0, 2);
9      let maxX = sl(8, 10);
10     let minX = maxX - sl(13, 18);
11     let X = [];
12     let Y = [];
13     for (let i = minX; i <= maxX; i += sl(1, 2))
14         X.push(i);
15     Y.push(sl(1, 6).pm());
16     for (let i = 1; i < X.length; i++) {
17         do {
18             if (i % 2)
19                 Y[i] = Y[i - 1] + sl(1, 4).pm();
20             else
21                 Y[i] = 0;
22         } while (Y[i].abs() > 5);
23     }
24     let spline = new Spline(X, Y);
25     let extremum = [];
26     let root = [];
27     let rootView = [];
28     for (let i = minX; i < maxX; i += 0.1) {
29         genAssert(f(i).abs() < 8, 'График заходит за рамки видимости');
30         if (f(i) < f(i - 0.1) && f(i) < f(i + 0.1) || (f(i) > f(i - 0.1) && f(i) >
31             f(i + 0.1))) {
32             extremum.push([i, f(i)]);
33         }
34         if (extremum.length > 0)
35             genAssert(extremum[extremum.length - 1][1].abs().round() != 0, 'Экстремум
находится слишком близко к оси Oх');
36     }
37     for (let i = minX; i < maxX; i += 0.001) {
38         if (f(i) * f(i - 0.001) < 0) {
39             root.push(i);
40             rootView.push(i.round());
41             genAssert((root[root.length - 1].abs() - rootView[rootView.length -
42                 1].abs()).abs() < 0.1);
43         }
44     }
45     genAssert(extremum.length > 2, 'Экстремумов недостаточно');
46
47     let maxLengthOfIncreasing = [];
48     let maxLengthOfDecreasing = [];
49
50     let interval = [minX].concat(rootView);
51     interval.push(maxX);
52
53     for (let i = 1; i < interval.length; i++) {
54         let len = interval[i] - interval[i - 1];
55         if (f(interval[i - 1] + len / 2) > 0)
56             maxLengthOfIncreasing.push(len);
57         else
58             maxLengthOfDecreasing.push(len);
59     }

```

```

60     chas2.task.modifiers.addCanvasIllustration({
61         width: 500,
62         height: 400,
63         paint: paint1,
64     });
65 });
66 })();

```

4. Отрисовываем график, определяем задание.

```

1  (function() {
2      retryWhileError(function() {
3          'use strict';
4
5          function f(x) {
6              return spline.at(x);
7          }
8          NAinfo.requireApiVersion(0, 2);
9          let maxX = sl(8, 10);
10         let minX = maxX - sl(13, 18);
11         let X = [];
12         let Y = [];
13         for (let i = minX; i <= maxX; i += sl(1, 2))
14             X.push(i);
15         Y.push(sl(1, 6).pm());
16         for (let i = 1; i < X.length; i++) {
17             do {
18                 if (i % 2)
19                     Y[i] = Y[i - 1] + sl(1, 4).pm();
20                 else
21                     Y[i] = 0;
22             } while (Y[i].abs() > 5);
23         }
24         let spline = new Spline(X, Y);
25         let extremum = [];
26         let root = [];
27         let rootView = [];
28         for (let i = minX; i < maxX; i += 0.1) {
29             genAssert(f(i).abs() < 8, 'Слишком большой горбик');
30             if (f(i) < f(i - 0.1) && f(i) < f(i + 0.1) || (f(i) > f(i - 0.1) && f(i) >
31 f(i + 0.1))) {
32                 extremum.push([i, f(i)]);
33             }
34             if (extremum.length > 0)
35                 genAssert(extremum[extremum.length - 1][1].abs().round() != 0, 'Слишком
непонятный экстремум');
36         }
37         for (let i = minX; i < maxX; i += 0.001) {
38             if (f(i) * f(i - 0.001) < 0) {
39                 root.push(i);
40                 rootView.push(i.round());
41                 genAssert((root[root.length - 1].abs() - rootView[rootView.length -
42 1].abs()).abs() < 0.1);
43             }
44         }
45         genAssert(extremum.length > 2, 'Экстремумов недостаточно');
46
47         let maxLengthOfIncreasing = [];
48         let maxLengthOfDecreasing = [];

```

```

48 let interval = [minX].concat(rootView);
49 interval.push(maxX);
50
51 for (let i = 1; i < interval.length; i++) {
52     let len = interval[i] - interval[i - 1];
53     if (f(interval[i - 1] + len / 2) > 0)
54         maxLengthOfIncreasing.push(len);
55     else
56         maxLengthOfDecreasing.push(len);
57 }
58
59
60 let condition = [['возрастания', [
61     ['наибольшего', maxLengthOfIncreasing.maxE()],
62     ['наименьшего', maxLengthOfIncreasing.minE()]
63 ].iz()],[ 'убывания', [
64     ['наибольшего', maxLengthOfDecreasing.maxE()],
65     ['наименьшего', maxLengthOfDecreasing.minE()]
66 ].iz()]].iz();
67
68 let paint1 = function(ct) {
69     let h = 380;
70     let w = 500;
71     ct.drawCoordinatePlane(w, h, {
72         hor: 1,
73         ver: 1
74     }, {
75         x1: '1',
76         y1: '1',
77         sh1: 13,
78     }, 20);
79     ct.font = "12px liberation_sans";
80     ct.drawLine(20 * maxX, 5, 20 * maxX, -5);
81     ct.drawLine(20 * minX, 5, 20 * minX, -5);
82     if (maxX != 0 && maxX != 1)
83         ct.fillText(maxX, 20 * maxX - 5, 15);
84     if (minX != 0 && minX != 1)
85         ct.fillText(minX, 20 * minX - 10, 15);
86     ct.scale(20, -20);
87     ct.lineWidth = 0.15;
88
89     graph9AdrawFunction(ct, f, {
90         minX: minX,
91         maxX: maxX,
92         minY: -9,
93         maxY: 9,
94         step: 0.01
95     });
96     graph9AmarkCircles(ct, [
97         [maxX, f(maxX)],
98         [minX, f(minX)]
99     ], 2, 0.2);
100     ct.fillStyle = "white";
101     graph9AmarkCircles(ct, [
102         [maxX, f(maxX)],
103         [minX, f(minX)]
104     ], 2, 0.1);
105
106     ct.fillStyle = "blue";
107 };

```

```

108     NAtask.setTask({
109         text: 'На рисунке изображен график производной функции  $f(x)$ , определенной на
интервале  $[' + \text{minX} + ';$  ' +
110              $\text{maxX} + ')$ . ' +
111             'Найдите промежутки '+condition[0]+' функции  $f(x)$ . В ответе укажите длину
'+condition[1][0]+' из них.',
112         answers: condition[1][1],
113     });
114     chas2.task.modifiers.addCanvasIllustration({
115         width: 500,
116         height: 400,
117         paint: paint1,
118     });
119 });
120 })();

```

1.2. Преимущества программной генерации заданий

На примере двух предыдущих задач были явно показано превосходство шаблонов над заданиями из Открытого Банка Заданий, а именно:

1. Большое количество разнообразных задач одного типа.
2. Простота и скорость написания шаблонов.
3. Невозможность нахождения учащимися ответов на задачи.

2. Глава вторая

2.1. Функции, используемые в проекте

За 9 лет работы над проектом «Час ЕГЭ» была разработана нестандартная библиотека для упрощения многих задач. Далее представлены наиболее используемые функции из неё.

Работа с массивами

Многочлены

Элементы одномерного массива `Array` — коэффициенты, стоящие в порядке возрастания степеней.

```
Array.prototype.mn_proizv = function()
```

Находит производную от многочлена.

```
Array.prototype.mn_vychisl = function()
```

Находит корни многочлена.

```
Array.prototype.mn_txt = function()
```

TeX-представление многочлена.

```
Array.prototype.mn_pervoobr = function()
```

Находит первообразную от многочлена.

```
Array.prototype.mn_txtmas = function()
```

TeX-представление многочлена.

```
Array.prototype.mt_pryam = function()
```

Возвращает коэффициенты a и b прямой $y = ax + b$, проходящей через две первые точки.

Вспомогательные функции для массивов

```
Array.prototype.shuffle = function(b)
```

Перемешивает массив случайным образом. Если `b`, то ещё и рекурсивно на один уровень.

```
Array.prototype.iz = function(p1)
```

Если `p1` опущено, возвращает случайный элемент массива, иначе массив `p1` неповторяющихся элементов массива.

Матрицы

```
function multiplyMatrix(A,B)
```

Умножает матрицу `A` на `B`, возвращает результат в матрице `C`.

```
function Determinant(A)
```

Возвращает определитель матрицы `A`.

```
function MatrixCofactor(i,j,A)
```

Возвращает алгебраическое дополнение матрицы A.

```
function AdjugateMatrix(A)
```

Возвращает союзную(присоединенную) матрицу.

```
function rang_mat(A)
```

Возвращает ранг матрицы A.

```
function InverseMatrix(B)
```

Возвращает обратную к B матрицу.

```
function generateMatrix(stroki, stolbcy, min, max, p1)
```

Генерирует матрицу из случайных чисел.

Работа с числами

```
Number.prototype.chislitlx = function(p1, p2)
```

Возвращает строку, состоящую из данного числа и подходящего падежа слова p1, при этом полученное словосочетанию стоит в падеже p2 (если не указан - именительный).

```
Number.prototype.pow = function(n)
```

Возвращает число в степени n.

```
Number.prototype.sqrt = function(n)
```

Возвращает квадратный корень из числа.

```
Number.prototype.sqr = function()
```

Возвращает квадрат числа.

```
Number.prototype.abs = function()
```

Возвращает модуль числа.

```
Number.prototype.floor = function()
```

Возвращает число, округленное до целого в меньшую сторону.

```
Number.prototype.ceil = function()
```

Возвращает число, округленное до целого в большую сторону.

```
Number.prototype.pm = function()
```

Случайным образом возвращает число или ему противоположное.

```
Number.prototype.ts = function()
```

Приводит число к стандартному виду (с десятичной запятой и не более чем 10 знаками после неё). Необходимо для отсе-
чения ложной точности.

Пример:

```
1 let number1=0.7+0.1;  
2 let number2=(0.7+0.1).ts()
```

number1 = 0.7999999999999999

number2 = 0,8

```
Number.prototype.texfracpi = function(p1)
```

Возвращает TeX-представление дроби, у которой в числителе данное число, умноженное на π , а в знаменателе p1. Случай p1 = 1 учитывается.

```
Number.prototype.texsqrt = function(p1, p2)
```

TeX-представление корня из данного числа. Если данное число - полный квадрат, то корень из числа. Если p1, то из-под корня будут вынесены возможные множители. Если p1, p2 и из-под корня выносится единица, то она будет опущена.

```
Number.prototype.isZ = function()
```

Возвращает true, если число n целое.

```
Number.prototype.isPolnKvadr = function()
```

Возвращает true, если число является полным квадратом.

2.2. Работа со строками

```
Number.prototype.toZagl = function()
```

Возвращает исходную строку с первой заглавной буквой.

```
Number.prototype.esli = function()
```

Возвращает данную строку, если p1, и пустую в противном случае.

```
Number.prototype.plusminus = function()
```

Возвращает упрощенное выражение, вставляя между числами необходимые знаки и убирая нулевые.

Работа с canvas

```
CanvasRenderingContext2D.prototype.drawLine = function(x1,y1,x2,y2)
```

Рисует линию из точки (x1,y1) в (x2,y2).

```
CanvasRenderingContext2D.prototype.fillKrug = function(x,y,r)
```

Рисует круг с центром в (x,y) и радиусом r.

```
CanvasRenderingContext2D.prototype.drawArrow
```

```
= function(x1, y1, x2, y2, arrowType)
```

Рисует стрелку из точки (x1,y1) в (x2,y2).

```
CanvasRenderingContext2D.prototype.drawCoordPlane
```

```
= function(w, h, kl, text, mash)
```

Рисует координатную плоскость. w и h – её размеры, kl – объект с полями hor и ver (высота и ширина клетки), text - объект с полями x1 и y1 (единичные отрезки типа string), sh1 и sh2 (шрифты для x1, y1, по умолчанию 12px) и mash - масштаб изображения (по умолчанию равно 1).

```
CanvasRenderingContext2D.prototype.setkaVer2
```

```
= function(h, w, hor, ver, mash)
```

Рисует прямоугольную сетку. w и h – её размеры, hor и ver – высота и ширина клетки, mash - масштаб (по умолчанию равно 1).

```
function graph9AmarkCircles(ct, XY, maxQuantity, radius)
```

Изображает не более maxQuantity точек в координатах из двумерно массива XY в виде кругов радиусом radius.

```
function graph9AdrawFunction(ct,f, o)
```

Рисует график f(x). Границы отображения задаются объектом o с полями maxX, maxY, minX, minY.

Вспомогательные функции

function sluchch(n,k,s)

Возвращает случайное число от n до k с шагом s (по умолчанию 1). Эта функция используется настолько часто, что для неё была придумана сокращённая форма sl().

function slKrome(a,p1,p2,p3)

Возвращает случайное число, кроме a. Если a – массив, то не содержащееся в нём; если число или строка, то не равное ему; Если функция, принимающая параметр - то не удовлетворяющее ей.

function sluchDel(a)

Возвращает случайный делитель числа a.

function sluchiz(a,n)

Возвращает массив из n случайных не повторяющихся элементов массива a.

function slLetter(b)

Возвращает случайную букву английского алфавита.

function intPoints(f,o)

Возвращает двумерный массив из всех целых точек графика f(x). Границы нахождения точек задаются объектом o с полями maxX, maxY, minX, minY.

2.3. Вклад автора в расширение каталога

Задание №109083

```
1 (function () {
2   'use strict';
3   NAinfo.requireApiVersion(0, 2);
4   let ves = sluchch(10, 100);
5   let pr1, pr2, answ;
6   do {
7     pr1 = sluchch(4, 20);
8     pr2 = sluchch(70, 90);
9     answ = (ves * (100 - pr1)) / (100 - pr2);
10  } while (!(answ * 100).isZ());
11  let pair = sklonlxkand([
12    ['абрикос', 'курага'],
13    ['виноград', 'изюм'],
14    [
15      ['финик', 'яблоко', 'груша', 'банан', 'дыня', 'персик',
16        'инжир', 'манго', 'папайя', 'хурма', 'ананас', 'кокос',
17        'шиповник', 'клюква', 'барбарис', 'клубника', 'вишня']
18    ].iz(),
19    'сухофрукт'
20  ],
21  ].iz());
22  NATask.setTask({
23    text: 'При сушке ' + pair[0].re + ' получается ' + pair[1].ie + '.' +
24    ' Сколько килограммов ' + pair[0].re + ' потребуется для получения'
25    + ves + ' килограммов ' + pair[1].re + ', ' +
26    'если ' + pair[0].ie + ' содержит '+pr2+ '% воды, а ' + pair[1].ie +
27    'содержит' + pr1 + '% воды?'
```

```

28         answers: answ,
29     });
30 }());
31 //Обзад 109083

```

Примеры генерируемых задач:

При сушке абрикоса получается курага. Сколько килограммов абрикоса потребуется для получения 66 килограммов кураги, если абрикос содержит 90% воды, а курага содержит 20% воды?

При сушке дыни получается сухофрукт. Сколько килограммов дыни потребуется для получения 73 килограммов сухофрукта, если дыня содержит 88% воды, а сухофрукт содержит 7% воды?

При сушке винограда получается изюм. Сколько килограммов винограда потребуется для получения 70 килограммов изюма, если виноград содержит 81% воды, а изюм содержит 5% воды?

Задание №99621

```

1  (function () {
2      'use strict';
3      NAinfo.requireApiVersion(0, 2);
4      let child_1 = sklonlxxkand(['Петя', 'Коля', 'Вася', 'Ваня', 'Саша', 'Женя', 'Никита',
5      'Арсений', 'Антон', 'Яков',
6      'Рома', 'Олег', 'Кирилл', 'Данил', 'Даниил', 'Арик', 'Ярик', 'Фома', 'Дима', 'Артём',
7      'Матвей', 'Максим', 'Игорь'
8      ].iz()),
9      child_2 = ['Алёна', 'Саша', 'Женя', 'Вика', 'Арина', 'Маша', 'Лена', 'Оля', 'Катя',
10     'Марина', 'Аня', 'Наташа',
11     'Ирина', 'Настя', 'Ирма', 'Кристина', 'Ира', 'Мила', 'Тома', 'Любовь3', 'Вера',
12     'Надежда', 'Снежана',
13     ].iz(),
14     work = sklonlxxkand(['работа', 'тест'].iz()),
15     nosame = 'одинаковый',
16     answ = ['выполнять ', 'решать ', 'делать '].iz();
17     let job, doing, kr = '';
18     if (work.ie == 'тест') {
19         job = sklonlxxkand('вопрос');
20         doing = 'отвечает';
21         answ = 'отвечать на ';
22     } else {
23         job = sklonlxxkand(['задание', 'задача', 'пример', 'номер'].iz());
24         kr = ['контрольную', 'проверочную', 'экзаменационную'].iz();
25         doing = ['решает', 'делает'].iz();
26     }
27     let v1 = sluchch(5, 9),
28         v2 = v1 + 1,
29         razn;
30     if (v1 == 7)

```

```

27     razn = [30, 60].iz();
28 else
29     razn = sluchch(20, 60, 10);
30 if (work.ie == 'работа')
31     nosame = 'одинаковую';
32 NATask.setTask({
33     text: ' ' + child_1.ie + ' и ' + child_2 + ' выполняют ' + nosame + ' ' +
kr.esli(work.ie == 'работа') + ' ' +
34     work.ve + ' . ' + child_2 + ' ' +
35     doing + ' за час ' + 'на'.esli(work.ie == 'тест') + ' ' + v1 + ' ' + job.rm
+ ' ' + work.re + ', а ' +
36     child_1.ie + ' ' +
37     'на'.esli(work.ie == 'тест ') + ' ' + v2 + ' . ' +
38     ' Они одновременно начали ' + answ + ' ' + job.vm + ', ' +
39     'и ' + child_2 + ' закончила ' + work.ve + ' позже ' + child_1.re + ' на
' + razn + ' ' +
40     'минут. Сколько ' + job.rm + ' содержит ' + work.ie + '?',
41     answers: (razn * v1 * v2) / (60 * (v2 - v1)),
42 });
43 })();
44 //Обзад 99621

```

Примеры генерируемых задач:

Максим и Арина выполняют одинаковую экзаменационную работу. Арина решает за час 7 заданий работы, а Максим 8. Они одновременно начали выполнять задания, и Арина закончила работу позже Максима на 60 минут. Сколько заданий содержит работа?

Матвей и Ирма выполняют одинаковую контрольную работу. Ирма решает за час 9 примеров работы, а Матвей 10. Они одновременно начали решать примеры, и Ирма закончила работу позже Матвея на 20 минут. Сколько примеров содержит работа?

Коля и Женя выполняют одинаковый тест. Женя отвечает за час на 9 вопросов теста, а Коля 10. Они одновременно начали отвечать на вопросы, и Женя закончила тест позже Коли на 40 минут. Сколько вопросов содержит тест?

Задание №26775

```

1 (function () {
2     'use strict';
3     NAinfo.requireApiVersion(0, 2);
4     let znam, chisl_1, chisl_2;
5     chisl_1 = Math.pow(sluchch(1, 5), 2);
6     chisl_2 = chisl_1 * Math.pow(sluchch(1, 10), 2);
7     znam = chisl_1 + chisl_2;
8
9     let ugol = om.latbukv.iz();

```

```

10 let k = sluchch(0, 100);
11 let graniz = [
12     ['${k.texfracpi(1)};${(k + 0.5).texfracpi(1)}'], 1, 1, 'I'],
13     ['${(k + 0.5).texfracpi(1)};${(k + 1).texfracpi(1)}'], -1, 2, 'II'],
14     ['${(k + 1).texfracpi(1)};${(k + 1.5).texfracpi(1)}'], 1, 3, 'III'],
15     ['${(k + 1.5).texfracpi(1)};${(k + 2).texfracpi(1)}'], -1, 4, 'VI'],
16 ].iz();
17 let zn_1, zn_2, zn_3, minus_1 = '',
18     minus_2 = '',
19     grelets = '';
20 switch (sluchch(0, 1)) { //выбираем исходные данные и что будем искать
21     case 0:
22         zn_1 = '\\tg';
23         zn_2 = '\\cos';
24         zn_3 = '\\sin';
25         if ((graniz[2] == 2) || (graniz[2] == 3)) // cos<0? Возможно, это можно както-
упростить.
26             minus_1 = '-';
27         if ((graniz[2] == 3) || (graniz[2] == 4)) { //sin<0?
28             grelets = '<';
29             minus_2 = '-';
30         } else
31             grelets = '>';
32         break;
33     case 1:
34         zn_1 = '\\ctg';
35         zn_2 = '\\sin';
36         zn_3 = '\\cos';
37         if ((graniz[2] == 3) || (graniz[2] == 4)) //sin<0?
38             minus_1 = '-';
39         if ((graniz[2] == 2) || (graniz[2] == 3)) // cos<0?
40             {
41                 minus_2 = '-';
42                 grelets = '<';
43             } else
44                 grelets = '>';
45         break;
46     }
47 let answ = graniz[1] * Math.sqrt(chisl_2 / chisl_1);
48 grelets = 'Угол( ${ugol} принадлежит ${graniz[3]} четверти -
${zn_3}${ugol})${grelets}0$)'
49
50 NATask.setTask({
51     text: 'Найдите ${zn_1}${ugol}$, если ${zn_2}${ugol}=${minus_1}
52     \\sqrt{ \\frac{ ${ chisl_1 } }{ ${ znam } } }$ и ${ugol} \\in ${ graniz[0] } $.',
53     answers: answ,
54     analys: 'Найдём ${zn_3}${ugol}=\\sqrt{1-(${minus_1}
55     \\sqrt{\\frac{ ${ chisl_1 } }{ ${ znam } }})^ { 2}}=${ minus_2 }
56     \\sqrt{ \\frac{ ${ chisl_2 } }{ ${ znam } } }${grelets}$. ' +
57     'Тогда ${zn_1}${ugol}=${' - '.если(!graniz[2] %
58     2)}\\sqrt{\\frac{ ${chisl_2} }{ ${chisl_1} }}=${answ}$',
59     });
60 }) ();
61 //Обзад 26775

```

Примеры генерируемых задач:

Найдите $\operatorname{ctg} H$, если $\sin H = -\sqrt{\frac{9}{738}}$ и $H \in (37; 752)$.

Найдём $\cos H \sqrt{1 - (-\sqrt{\frac{9}{738}})^2} = -\sqrt{\frac{729}{738}}$ (Угол H принадлежит III четверти - $\cos H < 0$). Тогда $\operatorname{ctg} H = \sqrt{\frac{729}{9}} = 9$

Найдите $\operatorname{tg} M$, если $\cos M = \sqrt{\frac{4}{260}}$ и $M \in (\frac{101}{2}; 51)$.

Найдём $\sin M = \sqrt{1 - (-\sqrt{\frac{9}{738}})^2} = -\sqrt{\frac{729}{738}}$ (Угол M принадлежит VI четверти - $\sin M < 0$). Тогда $\operatorname{tg} M = \sqrt{\frac{256}{4}} = -8$

Найдите $\operatorname{tg} Q$, если $\cos Q = -\sqrt{\frac{9}{153}}$ и $Q \in (\frac{171}{2}; 86)$.

Найдём $\sin Q = \sqrt{1 - (-\sqrt{\frac{9}{153}})^2} = \sqrt{\frac{144}{153}}$ (Угол Q принадлежит VI четверти - $\sin Q > 0$). Тогда $\operatorname{tg} Q = \sqrt{\frac{144}{9}} = -4$

Задание №26771

```

1 (function() {
2   'use strict';
3   NAinfo.requireApiVersion(0, 2);
4   let answ = sluchch(-100, 100, 0.5);
5   let ugol_1 = sluchch(1, 90);
6   let ugol_2 = ugol_1.pm() + 90 * sluchch(1, 11, 2);
7   ugol_1 += 90 * sluchch(0, 6, 2);
8   let k_1 = sluchch(0, 1);
9   let k_2 = 1 - k_1;
10
11   let zn = ['\\tg', '\\ctg'];
12
13   NATask.setTask({
14     text: 'Найдите значение выражения $' + answ + ' + zn[k_1] + ' + ugol_1 + '^\\circ \\cdot ' + zn[k_1] + ' + ugol_2 + '^\\circ.$',
15     answers: answ,
16     analys: 'По формуле приведения $' + zn[k_1] + ' + ugol_2 + '^\\circ=' + zn[k_1] + '(' + (ugol_1 + ugol_2) + '^\\circ-' + ugol_2 + '^\\circ)=' + zn[k_2] + ugol_1 + '^\\circ.$' + ' Тогда $' + answ + ' + zn[k_1] + ' + ugol_1 + '^\\circ \\cdot ' + zn[k_2] + ' + ugol_1 + '^\\circ=' + answ + '$',
17   });
18 });
19 //Обзад 26771 26770

```

Примеры генерируемых задач:

Найдите значение выражения $-72 \operatorname{tg} 201^\circ \cdot \operatorname{tg} 249^\circ$.

По формуле приведения $\operatorname{tg} 249^\circ = \operatorname{tg}(450^\circ - 249^\circ) = \operatorname{ctg} 201^\circ$. Тогда $-72 \operatorname{tg} 201^\circ \cdot \operatorname{ctg} 201^\circ = -72$

Найдите значение выражения $-58 \operatorname{tg} 400^\circ \cdot \operatorname{tg} 410^\circ$.

По формуле приведения $\operatorname{tg} 410^\circ = \operatorname{tg}(810^\circ - 410^\circ) = \operatorname{ctg} 400^\circ$. Тогда $-58 \operatorname{tg} 400^\circ \cdot \operatorname{ctg} 400^\circ = -58$

Найдите значение выражения $-32 \operatorname{ctg} 195^\circ \cdot \operatorname{ctg} 285^\circ$.

По формуле приведения $\operatorname{ctg} 285^\circ = \operatorname{ctg}(480^\circ - 285^\circ) = \operatorname{tg} 195^\circ$. Тогда $-32 \operatorname{ctg} 195^\circ \cdot \operatorname{tg} 195^\circ = -32$

Задание №320169

```
1 (function () {
2   'use strict';
3   NAinfo.requireApiVersion(0, 0);
4   let kol = [[4, 'четырёх'], [5, 'пяти'], [8, 'восьми']].iz();
5   let chislo = kol.pop();
6   let k = sluchch(1, kol - 1);
7   let person1 = ['Петя', 'Коля', 'Вася', 'Ваня', 'Никита', 'Арсений', 'Антон', 'Яков',
8     'Рома', 'Олег', 'Кирилл', 'Данил', 'Даниил', 'Арик', 'Ярик', 'Фома', 'Дима', 'Артём',
9     'Матвей', 'Максим', 'Игорь',
10    ].iz(k);
11   let person2 = ['Алёна', 'Вика', 'Арина', 'Маша', 'Лена', 'Оля', 'Катя', 'Марина', 'Аня',
12     'Натasha',
13     'Ирина', 'Настя', 'Ирма', 'Кристина', 'Ира', 'Мила', 'Тома', 'Любовь', 'Вера',
14     'Надежда', 'Снежана',
15    ].iz(kol - k);
16   let persons = person1.concat(person2).shuffle();
17   let uslov = [persons.iz(), 'девочка', 'мальчик',].iz();
18   let must = 'должен';
19   if (person2.includes(uslov) || uslov == 'девочка')
20     must = 'должна';
21   let answ;
22   switch (uslov) {
23     case ('мальчик'):
24       answ = k / kol;
25       break;
26     case ('девочка'):
27       answ = (kol - k) / kol;
28       break;
29     default:
30       answ = 1 / kol;
31       break;
32   }
33   if (sl1()) {
34     uslov = 'не ' + uslov;
35     answ = 1 - answ;
36   }
37   let children = '';
38   for (let i = 0; i < kol - 1; i++) {
39     children += persons[i];
40     if (i != kol - 2)
41       children += ', ';
42   }
43 }
```

```

39     else
40         children += ' ';
41     }
42     children += 'и ' + persons[kol - 1];
43     NATask.setTask({
44         text: ' ' + children + ' бросили жребий - кому начинать игру. Найдите вероятность того,
что начинать игру ' + must + ' будет ' + uslov + '.',
45         answers: answ,
46         analys: 'Жребий начать игру может выпасть каждому из ' + chislo + ' детей. Вероятность
того, что это будет ' + uslov + ', равна ' + answ + '.',
47     });
48
49 })();
50 // 320169,320335,320343

```

Примеры генерируемых задач:

Марина, Вика, Коля, Даниил, Арик, Надежда, Снежана и Наташа бросили жребий - кому начинать игру. Найдите вероятность того, что начинать игру должен будет мальчик.
Жребий начать игру может выпасть каждому из восьми детей. Вероятность того, что это будет мальчик, равна 0.375.

Антон, Олег, Ира, Артём и Коля бросили жребий - кому начинать игру. Найдите вероятность того, что начинать игру должна будет не девочка.
Жребий начать игру может выпасть каждому из пяти детей. Вероятность того, что это будет не девочка, равна 0.8.

Артём, Ирма, Данил, Антон и Максим бросили жребий - кому начинать игру. Найдите вероятность того, что начинать игру должна будет Ирма.
Жребий начать игру может выпасть каждому из пяти детей. Вероятность того, что это будет Ирма, равна 0.2.

Задание №320207

```

1  (function() {
2      'use strict';
3      NAinfo.requireApiVersion(0, 0);
4      let ill = sklonlykand(['гепатит', 'криворучие', 'соннис', 'забыватикус', 'переленеус',
'шоколадодифицит',
5          'антилогизм',
6          'диабет', 'Ёжиковирус-', 'перфектолиз', 'трудоголизм', 'хладоручие', 'жмотство',
'физикозависимость-',
7          'Фентезикоз',
8      ]).iz();
9      let persons = sklonlykand(['физик', 'химик', 'школьник', 'чудик', 'чиновник', 'президент',
'пациент', 'студент',
10         'математик',
11     ]).iz();
12     let realRez, lozhRez, realIll, answ, prAnsw1, prAnsw2;
13     do {

```

```

14     realRez = sluchch(30, 95);
15     lozhRez = sluchch(2, 10);
16     realIll = sluchch(1, 95);
17     prAnsw1 = (realRez * realIll) / 10000;
18     prAnsw2 = (lozhRez * (100 - realIll)) / 10000;
19     answ = (realRez * realIll + lozhRez * (100 - realIll)) / 10000;
20 } while (!(answ * 10000.0).isZ());
21 let Rez;
22 if (sl1())
23     Rez = 'положительным';
24 else {
25     Rez = 'отрицательным';
26     answ = 1 - answ;
27 }
28 let veroyt = ['с вероятностью ', 'в '];
29 if (sl1())
30     realRez /= 100;
31 if (sl1())
32     lozhRez /= 100;
33 if (sl1())
34     realIll /= 100;
35
36 let find = ['вероятность', 'процент'];
37 if (sl1())
38     find.pop();
39 else {
40     find.shift();
41     answ *= 100;
42 }
43 NAtask.setTask({
44     text: ' Всем ${persons.dm} с подозрением на ${ill.ve} делают анализ крови. ' +
45         'Если анализ выявляет ${ill.ve}, то результат анализа называется положительным. ' +
46         'У больных ${ill.te} ${persons.rm} анализ даёт положительный результат
47     ${veroyt[0].esli(realRez<1)} ${veroyt[1].esli(realRez>1)} ${realRez.ts()}' +
48     ' ${esli(realRez>1)}. ' +
49     'Если ${persons.im} не болен ${ill.te}, то анализ может дать ложный положительный
50     результат ${veroyt[0].esli(lozhRez<1)} ${veroyt[1].esli(lozhRez>1)}
51     ${lozhRez.ts()}' +
52     ' ${esli(lozhRez>1)}. ' +
53     'Известно, что ${realIll.ts()}' +
54     ' ${esli(realIll>1)} ${persons.rm}, поступающих с подозрением на ${ill.ve},
55     действительно больны ${ill.te}. ' +
56     'Найдите ${find} того, что результат анализа у ${persons.rm}, поступившего в
57     клинику с подозрением на ${ill.ve}, будет ${Rez}.',
58     answers: answ,
59     analys: 'Анализ ${persons.rm} может быть положительным по двум причинам: ' +
60         ' 1) ${persons.ie} болеет ${ill.te}, его анализ верен; 2) пациент не болеет
61         ${ill.te}, его анализ ложен. ' +
62         ' По формуле условной вероятности: $$$realRez.ts()$$$ / 100
63         '.esli(realRez>1)}\cdot${realIll.ts()}' / 100 '.esli(realIll>1)} =
64         ${prAnsw1.ts()}' +
65         ' и $$$lozhRez.ts()$$$ / 100 '.esli(lozhRez>1)}\cdot(1-${realIll.ts()}' /
66         100 '.esli(realIll>1)}) = ${prAnsw2.ts()}' +
67         'События быть больным или быть здоровым образуют полную группу они( несовместны и одно
68         из них непременно наступает), ' +
69         ' поэтому можно применить формулу полной вероятности. Тогда
70         $$$prAnsw1.ts()+${prAnsw2.ts()}=${(prAnsw1+prAnsw2).ts()}' +
71         '${'
72     Необходимо найти,
73     что анализ будет отрицательным: $1 - ${

```



```

63         (prAnsw1 + prAnsw2).ts()
64     }
65     $.
66     '.esli(Rez == '
67     отрицательным ')} ' +
68     '${'
69     Так как нужно найти процент,
70     умножим полученный ответ на 100: $$ {
71         (prAnsw1 + prAnsw2).ts()
72     } \ \
73     cdot 100 = $ {
74         answ.ts()
75     }
76     $ '.esli(find == '
77     процент ')} ' +
78     'Ответ: $$ {answ.ts()} $ ',
79 });
80
81 }) ();
82 // 320207

```

Примеры генерируемых задач:

Всем школьникам с подозрением на жмотство делают анализ крови. Если анализ выявляет жмотство, то результат анализа называется положительным. У больных жмотством школьников анализ даёт положительный результат в 66%. Если школьники не болен жмотством, то анализ может дать ложный положительный результат с вероятностью 0,09. Известно, что 0,4 школьников, поступающих с подозрением на жмотство, действительно больны жмотством. Найдите вероятность того, что результат анализа у школьников, поступившего в клинику с подозрением на жмотство, будет положительным.

Анализ школьников может быть положительным по двум причинам: 1) школьник болеет жмотством, его анализ верен; 2) пациент не болеет жмотством, его анализ ложен. По формуле условной вероятности: $66/100 \cdot 0,4 = 0,264$ и $0,09 \cdot (1 - 0,4) = 0,054$. События быть больным или быть здоровым образуют полную группу (они несовместны и одно из них непременно наступает), поэтому можно применить формулу полной вероятности. Тогда $0,264 + 0,054 = 0,318$. Ответ: 0,318

Всем чиновникам с подозрением на диабет делают анализ крови. Если анализ выявляет диабет, то результат анализа называется положительным. У больных диабетом чиновников анализ даёт положительный результат с вероятностью 0,67. Если чиновники не болен диабетом, то анализ может дать ложный положительный результат в 9%. Известно, что 73% чиновников, поступающих с подозрением на диабет, действительно больны диабетом. Найдите процент того, что результат анализа у чиновников, поступившего в клинику с подозрением на диабет, будет отрицательным.

Анализ чиновников может быть положительным по двум причинам: 1) чиновник болеет диабетом, его анализ верен; 2) пациент не болеет диабетом, его анализ ложен. По формуле условной вероятности: $0,67 \cdot 73/100 = 0,4891$ и $9/100 \cdot (1 - 73/100) = 0,0243$. События быть больным или быть здоровым образуют полную группу (они несовместны и одно из них непременно наступает), поэтому можно применить формулу полной вероятности. Тогда $0,4891 + 0,0243 = 0,5134$. Необходимо найти, что анализ будет отрицательным: $1 - 0,5134$. Так как нужно найти процент, умножим полученный ответ на 100: $0,5134 \cdot 100 = 48,66$ Ответ: 48,66

Всем школьникам с подозрением на перфектолиз делают анализ крови. Если анализ выявляет перфектолиз, то результат анализа называется положительным. У больных перфектолизом школьников анализ даёт положительный результат в 91%. Если школьники не болен перфектолизом, то анализ может дать ложный положительный результат с вероятностью 0,09. Известно, что 0,47 школьников, поступающих с подозрением на перфектолиз, действительно больны перфектолизом. Найдите процент того, что результат анализа у школьников, поступившего в клинику с подозрением на перфектолиз, будет отрицательным.

Анализ школьников может быть положительным по двум причинам: 1) школьник болеет перфектолизом, его анализ верен; 2) пациент не болеет перфектолизом, его анализ ложен. По формуле условной вероятности: $91/100 \cdot 0,47 = 0,4277$ и $0,09 \cdot (1 - 0,47) = 0,0477$. События быть больным или быть здоровым образуют полную группу (они несовместны и одно из них непременно наступает), поэтому можно применить формулу полной вероятности. Тогда $0,4277 + 0,0477 = 0,4754$. Необходимо найти, что анализ будет отрицательным: $1 - 0,4754$. Так как нужно найти процент, умножим полученный ответ на 100: $0,4754 \cdot 100 = 47,54$ Ответ: 47,54

Задание №50927103

```
1 retryWhileUndefined(function () {  
2     NAinfo.requireApiVersion(0, 2);  
3 }
```

```

4  function fp(x) {
5      return k * x + d;
6  }
7
8  function fs(x) {
9      return a * (x + b).sqrt() + c;
10 }
11
12 let x1 = sluchch(1, 6).pm();
13 let x2 = sluchch(8, 20);
14 let y1 = sl(-6, 6);
15 let y2 = sl(0, 20).pm();
16 let k = (y1 - y2) / (x1 - x2);
17 if (!(k * 1000).isZ() || !k)
18     return;
19 let d = y1 - k * x1;
20 let b = slKrome([-x1, -x2].iz(), -6, 6, 0.5);
21 let a = (y1 - y2) / ((x1 + b).sqrt() - (x2 + b).sqrt());
22 if (!(a * 1000).isZ())
23     return;
24 let c = y1 - a * (x1 + b).sqrt();
25 let pointss = intPoints(fs, {
26     minX: -5,
27     maxX: 5,
28     minY: -5.5,
29     maxY: 5.5,
30     step: 1,
31 });
32 if (pointss.length < 3)
33     return;
34 let pointsp = intPoints(fp, {
35     minX: -5,
36     maxX: 5,
37     minY: -5.5,
38     maxY: 5.5,
39     step: 1,
40 });
41 if (pointsp.length < 2)
42     return;
43 let find, answ;
44 if (sl1()) {
45     answ = x2;
46     find = 'абсциссу';
47 } else {
48     answ = y2;
49     find = 'ординату';
50 }
51 let paint1 = function (ct) {
52     let h = 300;
53     let w = 300;
54     //Оси координат
55     ct.drawCoordPlane(w, h, {
56         hor: 1,
57         ver: 1
58     }, {
59         x1: '1',
60         y1: '1',
61         sh1: 13,
62     }, 20);
63     //График

```

```

64     ct.scale(20, -20);
65     ct.lineWidth = 0.1;
66     graph9AdrawFunction(ct, fs, {
67         minX: -6.5,
68         maxX: 7,
69         minY: -7,
70         maxY: 6,
71         step: 0.0005,
72     });
73     graph9AdrawFunction(ct, fp, {
74         minX: -6.5,
75         maxX: 7,
76         minY: -7,
77         maxY: 6,
78         step: 0.05,
79     });
80     //точки
81     graph9AmarkCircles(ct, pointss, 3, 0.15);
82     graph9AmarkCircles(ct, pointsp, 2, 0.15);
83     graph9AmarkCircles(ct, [
84         [x1, y1]
85     ], 2, 0.15);
86     //буква
87     ct.fillStyle = "blue";
88     ct.font = "18px liberation_sans";
89     ct.scale(1 / 20, -1 / 20);
90     ct.fillText('A', 20 * x1 - 10, -20 * y1 - 10);
91 };
92 NAtask.setTask({
93     text: 'На рисунке изображены графики функций  $f(x)=a\sqrt{x} + [ '+', '- ' ].iz() +$   

94     ' $b+c$  и  $g(x)=kx + [ '+', '- ' ]$   

95     ' $d$ , ' +  

96     ' которые пересекаются в точках  $A$  и  $B$ . Найдите ' + find + ' точки  $B$ .',
97     answers: answ,
98     analys: ' $f(x)=$  ' + (a + ' $\sqrt{x} + b + [ '+', '- ' ] + c + '$').plusminus() + '<br>' +$   

99     ' $g(x)=$  ' + (k.ts() + ' $x + d$ ').plusminus() + ' $<br>' +$   

100     ' $A($  ' + x1 + ' $;$  ' + y1 + ' $)<br>' +$   

101     ' $B($  ' + x2 + ' $;$  ' + y2 + ' $)$'$ ,
102 });
103 chas2.task.modifiers.addCanvasIllustration({
104     width: 300,
105     height: 300,
106     paint: paint1,
107 });
108 return true;
109 }, 100000);
110 //50927103

```

Примеры генерируемых задач:

На рисунке изображены графики функций $f(x) = a\sqrt{x-b} + c$ и $g(x) = kx + d$, которые пересекаются в точках А и В. Найдите абсциссу точки В.

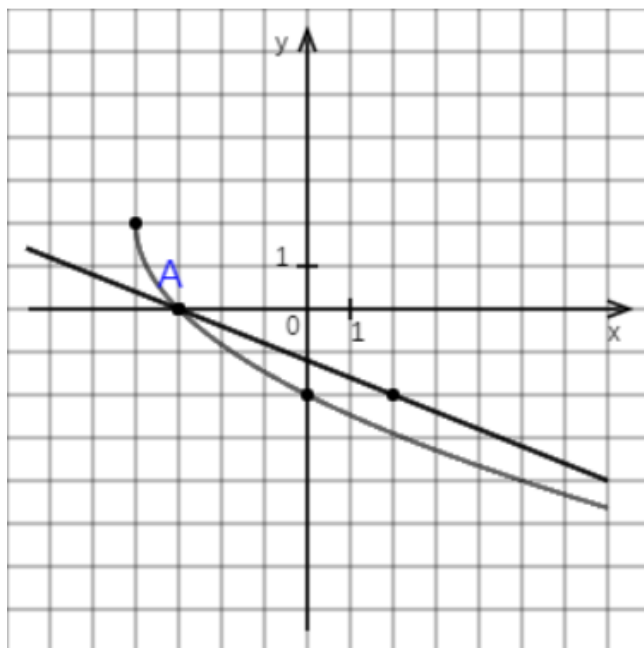
Решение:

$$f(x) = -2\sqrt{x+4} + 2$$

$$g(x) = -0,4x - 1,2$$

$$A(-3; 0)$$

$$B(12; -6)$$



На рисунке изображены графики функций $f(x) = a\sqrt{x-b} + c$ и $g(x) = kx - d$, которые пересекаются в точках А и В. Найдите абсциссу точки В.

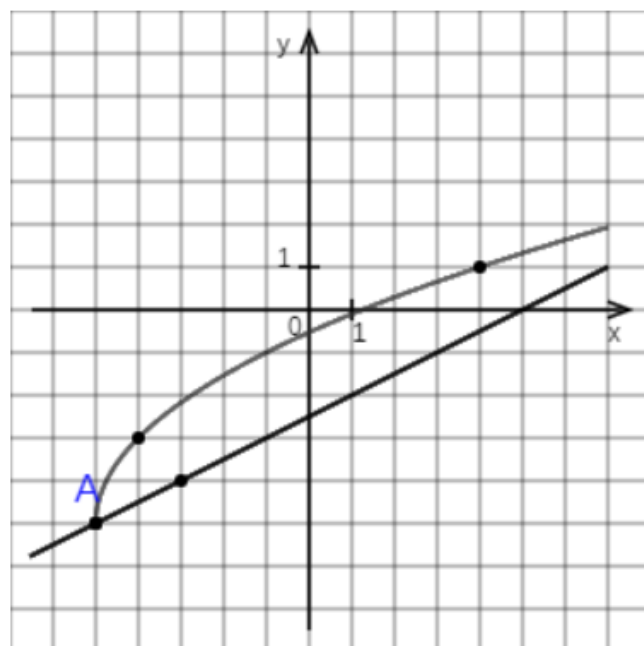
Решение:

$$f(x) = 2\sqrt{x+5} - 5$$

$$g(x) = 0,5x - 2,5$$

$$A(-5; -5)$$

$$B(11; 3)$$



На рисунке изображены графики функций $f(x) = a\sqrt{x-b} + c$ и $g(x) = kx - d$, которые пересекаются в точках А и В. Найдите ординату точки В.

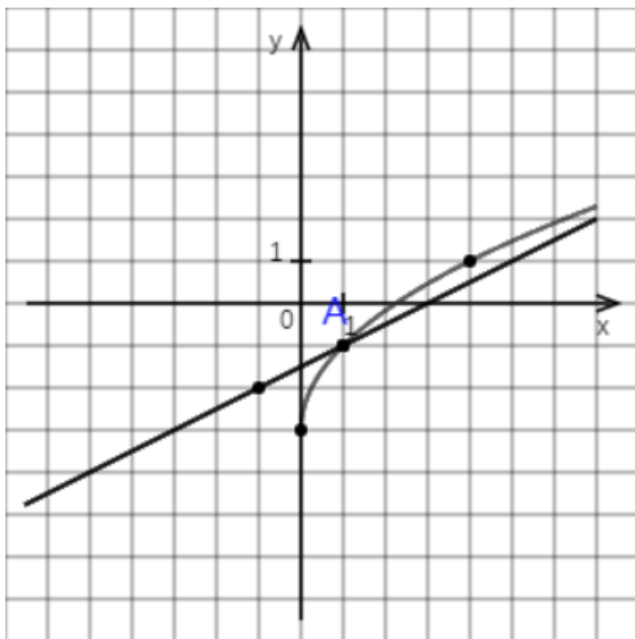
Решение:

$$f(x) = 2\sqrt{x+0} - 3$$

$$g(x) = 0,5x - 1,5$$

$$A(1; -1)$$

$$B(9; 3)$$



Задание №3509123

```

1  retryWhileUndefined(function () {
2      NAinfo.requireApiVersion(0, 2);
3
4      function variant(a, b, x) {
5          switch (trigfuncs) {
6              case 'sin':
7                  return a * Math.sin(x * Math.PI / 2) + b;
8              case 'cos':
9                  return a * Math.cos(x * Math.PI / 2) + b;
10             case 'tg':
11                 return a * Math.tan(x * Math.PI / 2) + b;
12             case 'ctg':
13                 return a * (1 / Math.tan(x * Math.PI / 2)) + b;
14             }
15         }
16
17         function f(x) {
18             return 0.5 * variant(a, b, x);
19         }
20         let trigfuncs = ['sin', 'cos', 'tg', 'ctg'].iz();
21         let a = sluchch(1, 6).pm();
22         let b = sluchch(0, a).pm();
23         let find, answ;
24         if (sl1() && trigfuncs != 'tg' && trigfuncs != 'ctg') {
25             find = 'a';
26             answ = a;
27         } else {
28             find = 'b';
29             answ = b;
30         }
31         let X = [],
32             Y = [];
33         for (let i = -2; i < 4; i++)
34             if (2 * f(i) < 5 && 2 * f(i) > -6) {
35                 X.push(i);
36                 Y.push(f(i));

```

```

37     }
38     if (X.length < 2)
39         return;
40     let paint1 = function (ct) {
41         let h = 300;
42         let w = 300;
43         //Оси координат
44         ct.drawCoordPlane(w, h, {
45             hor: 2,
46             ver: 1
47         }, {
48             x1: 'π/2',
49             y1: '1',
50             sh1: 13,
51         }, 20);
52         //График
53         ct.scale(40, -40);
54         ct.lineWidth = 0.05;
55         graph9AdrawFunction(ct, f, {
56             minX: -2.6,
57             maxX: 4,
58             minY: -4,
59             maxY: 3,
60             step: 0.05,
61         });
62         //точки
63         graph9AmarkCircles(ct, [X, Y].T(), 2, 0.1);
64     };
65     NATask.setTask({
66         text: 'На рисунке изображён график функции $f(x)=a\\' + trigfuncs + ' x+b$. Найдите
67 $' + find + '$.',
68         answers: answ,
69         analys: '$f(x)=' + (a + '\\\' + trigfuncs + ' x+' + b).replace('+0',
70 '').plusminus() + '$',
71     });
72     chas2.task.modifiers.addCanvasIllustration({
73         width: 300,
74         height: 300,
75         paint: paint1,
76     });
77     return true;
78 }, 100000);
79 //509123 509130 509137 509143 509287 509295

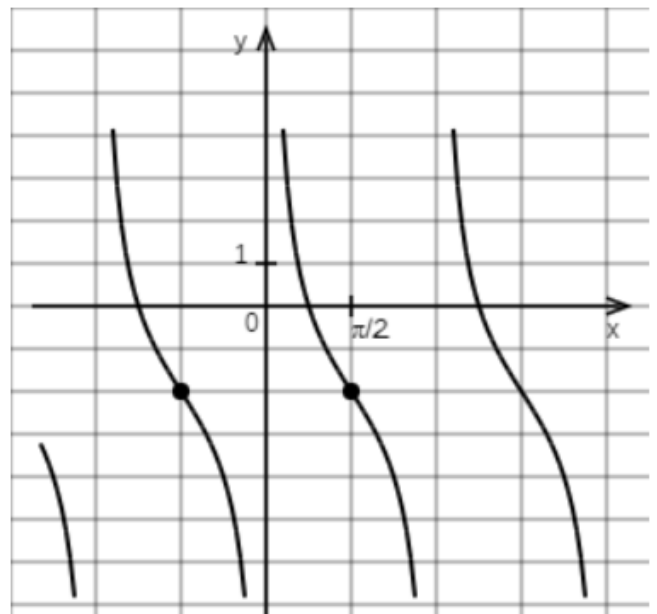
```

Примеры генерируемых задач:

На рисунке изображён график функции $f(x) = a \operatorname{ctg} x + b$. Найдите b .

Решение:

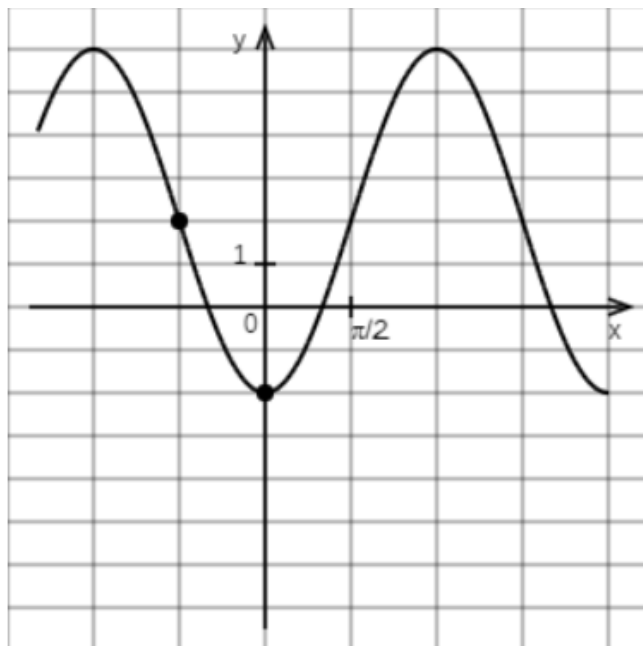
$$f(x) = 2 \operatorname{ctg} x - 2$$



На рисунке изображён график функции $f(x) = a \cos x + b$. Найдите a .

Решение:

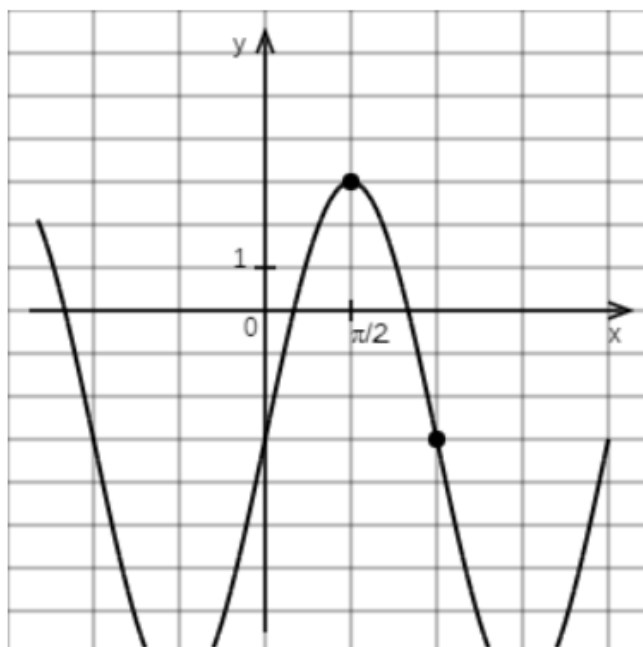
$$f(x) = -4 \cos x + 2$$



На рисунке изображён график функции $f(x) = a \sin x + b$. Найдите b .

Решение:

$$f(x) = 6 \sin x - 3$$



Задание №317541

```

1 (function() {
2   retryWhileError(function() {
3     'use strict';
4
5     function f(x) {
6       return spline.at(x);
7     }
8     NAinfo.requireApiVersion(0, 2);
9     let maxX = sl(5, 10);
10    let minX = maxX - sl(10, 15);
11    let X = [];
12    let Y = [];
13    for (let i = minX; i <= maxX; i += sl(1.5, 4, 0.1))
14      X.push(i);

```



```

15 Y.push(sl(1, 6).pm());
16 for (let i = 1; i < X.length; i++) {
17     do {
18         Y[i] = Y[i - 1] + sl(2, 10).pm();
19     } while (Y[i].abs() > 5 || Y[i] == 0);
20 }
21 let spline = new Spline(X, Y);
22 let extremum = [];
23
24 for (let i = minX; i < maxX; i += 0.1) {
25     genAssert(f(i).abs() < 8, 'Слишком большой горбик');
26     if (f(i) < f(i - 0.1) && f(i) < f(i + 0.1) || (f(i) > f(i - 0.1) && f(i) > f(i +
27 0.1))) {
28         extremum.push(f(i));
29     }
30 }
31 genAssert(extremum.length > 2, 'Максимумов недостаточно');
32
33 let funcAscendingPoints = [];
34 let funcDescendingPoints = [];
35 let points = [];
36 for (let i = minX + 1; i < maxX; i += sl(1, 3, 0.1)) {
37     if (f(i) > 0) {
38         genAssert(f(i).abs() > 1.5, 'Точка слишком близко к 0');
39         points.push([i, f(i)]);
40         funcAscendingPoints.push(i);
41     } else {
42         genAssert(f(i).abs() > 1.5, 'Точка слишком близко к 0');
43         points.push([i, f(i)]);
44         funcDescendingPoints.push(i);
45     }
46 }
47
48 genAssert(points.length > 3, 'точек мало');
49 let condition = [
50     ['возрастания', funcAscendingPoints],
51     ['убывания', funcDescendingPoints]
52 ].iz();
53 let paint1 = function(ct) {
54     let h = 380;
55     let w = 500;
56     ct.drawCoordinatePlane(w, h, {
57         hor: 1,
58         ver: 1
59     }, {
60         x1: '',
61         y1: '',
62         sh1: 17,
63     }, 40);
64
65     ct.fillStyle = "black";
66     ct.scale(20, -20);
67     ct.lineWidth = 0.15;
68
69     graph9AdrawFunction(ct, f, {
70         minX: minX,
71         maxX: maxX,
72         minY: -9,
73         maxY: 9,
74         step: 0.01

```

```

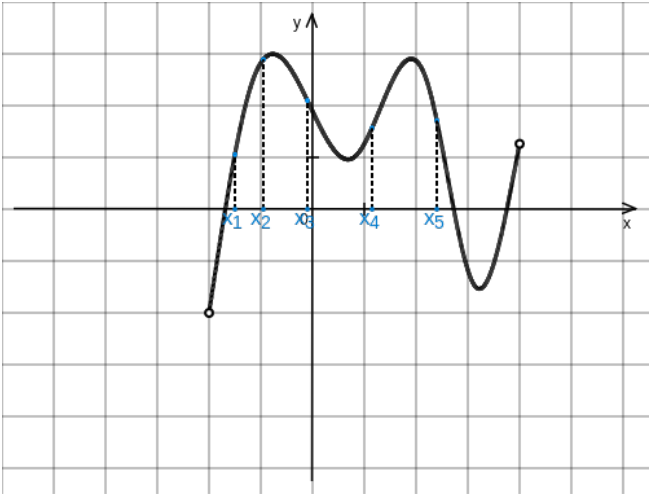
74     });
75     graph9AmarkCircles(ct, [
76         [maxX, f(maxX)],
77         [minX, f(minX)]
78     ], 2, 0.2);
79     ct.fillStyle = "white";
80     graph9AmarkCircles(ct, [
81         [maxX, f(maxX)],
82         [minX, f(minX)]
83     ], 2, 0.1);
84
85     ct.fillStyle = "#0099ff";
86     graph9AmarkCircles(ct, points, points.length, 0.1);
87
88
89     ct.scale(1 / 20, -1 / 20);
90     ct.lineWidth = 2;
91     ct.fillStyle = "#007acc";
92     for (let i = 0; i < points.length; i++) {
93         ct.font = "17px liberation_sans";
94
95         if (f(points[i][0]) < 0) {
96             ct.fillText('x', points[i][0] * 20, -3);
97             ct.font = "14px liberation_sans";
98             ct.fillText((i + 1).toString(), (points[i][0] + 0.4) * 20, 0);
99         } else {
100             ct.fillText('x', (points[i][0] - 0.5) * 20, 12);
101             ct.font = "14px liberation_sans";
102             ct.fillText((i + 1).toString(), (points[i][0] - 0.1) * 20, 15);
103         }
104
105         ct.setLineDash([4, 2]);
106         ct.drawLine(points[i][0] * 20, 0, points[i][0] * 20, -points[i][1] * 20);
107         graph9AmarkCircles(ct, [
108             [points[i][0] * 20, 0]
109         ], 1, 2);
110
111     }
112
113 };
114 NATask.setTask({
115     text: 'На рисунке изображён график  $y=f'(x)$  - производной функции  $f(x)$ . ' +
116         'На оси абсцисс отмечены ' + chislitlx(points.length, 'точка') + ':  $x_1$ ,  $x_2$ ,  

 $x_3$ , ...  $x_n$  + points.length +
117         ' $f(x)$ . Сколько из этих точек лежит на промежутках ' + condition[0] + ' функции  $f(x)$ ?',
118     answers: condition[1].length,
119 });
120 chas2.task.modifiers.addCanvasIllustration({
121     width: 500,
122     height: 400,
123     paint: paint1,
124 });
125 });
126 })();
127 //SugarHedgehog
128 //317541 317745 513618 516292 516325 628743 628769 317747 317749 317751 317753
129 //317755 317757 317759 317761 317763 317765 317767 317769 317771 317773 317775
130 //317777 317779 317781 317783 317785 317787 317789 317791 317793 317795 317797
131 //317799 317801 317803 317805 317807 317809 317811 317813 317815 317817 317819
132 //317821 317823 317825 317827 317829 317831 317833 317835 317837 317839 317841

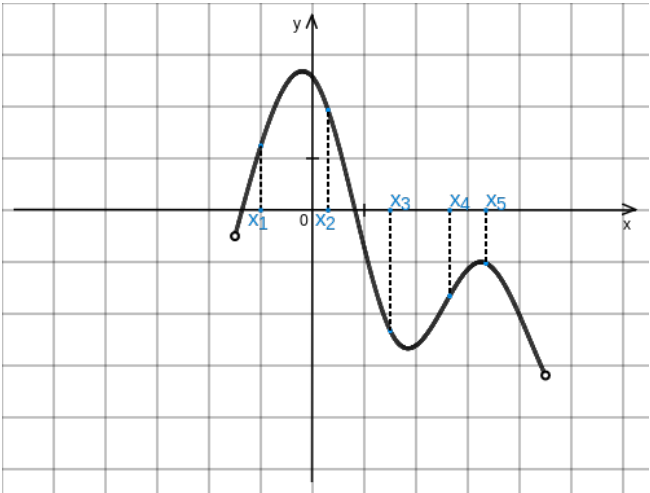
```

Примеры генерируемых задач:

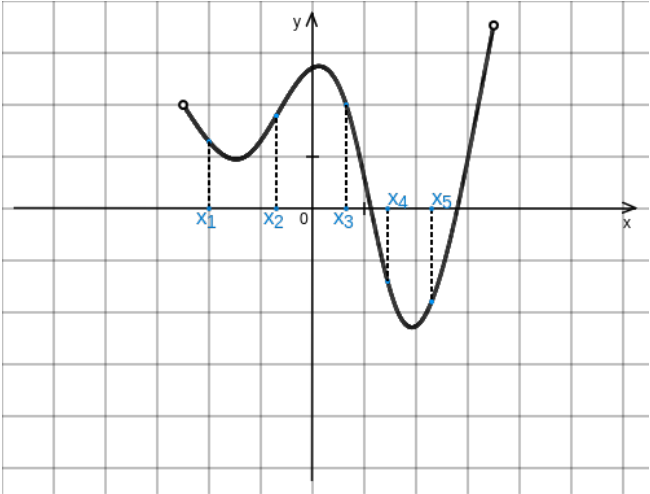
На рисунке изображён график дифференцируемой функции $y = f(x)$. На оси абсцисс отмечены 5 точек : $x_1, x_2, x_3, \dots, x_5$. Среди этих точек найдите все точки, в которых производная функции $f(x)$ положительна. В ответе укажите количество найденных точек.



На рисунке изображён график $y = f'(x)$ — производной функции $f(x)$. На оси абсцисс отмечены 5 точек: $x_1, x_2, x_3, \dots, x_5$. Сколько из этих точек лежит на промежутках возрастания функции $f(x)$?



На рисунке изображён график дифференцируемой функции $y = f(x)$. На оси абсцисс отмечены 5 точек : $x_1, x_2, x_3, \dots, x_5$. Среди этих точек найдите все точки, в которых производная функции $f(x)$ положительна. В ответе укажите количество найденных точек.



Задание №317544

```
1 (function() {
2   retryWhileError(function() {
3     'use strict';
4
5     function f(x) {
6       return spline.at(x);
7     }
8     NAinfo.requireApiVersion(0, 2);
9     let maxX = 10;
10    let minX = -10;
11    let X = [];
12    let Y = [];
13    for (let i = minX; i <= maxX; i += sl(1.5, 4, 0.1))
14      X.push(i);
15    Y.push(sl(1, 6).pm());
16    for (let i = 1; i < X.length; i++) {
17      do {
18        Y[i] = Y[i - 1] + sl(2, 10).pm();
19      } while (Y[i].abs() > 5 || Y[i] == 0);
20    }
21    let spline = new Spline(X, Y);
22    let extremum = [];
23
24    for (let i = minX; i < maxX; i += 0.1) {
25      genAssert(f(i).abs() < 8, 'Слишком большой горбик');
26      if (f(i) < f(i - 0.1) && f(i) < f(i + 0.1) || (f(i) > f(i - 0.1) && f(i) > f(i +
27        0.1))) {
28        extremum.push(f(i));
29      }
30    }
31    genAssert(extremum.length > 2, 'Максимумов недостаточно');
32
33    let funcAscendingPoints = [];
34    let funcDescendingPoints = [];
35
36    for (let i = minX + 1; i < maxX; i += sl(2, 6, 0.1)) {
37      let step = 0.1;
38      if (f(i) > f(i - step) && f(i) < f(i + step)) {
39        if ((f(i) > f(i - step) && f(i) < f(i + step)))
40          genAssert(f(i).abs() > 1.5, 'Точка слишком близко к 0');
41        funcAscendingPoints.push([i, f(i)]);
42      }
43      if (f(i) < f(i - step) && f(i) > f(i + step))
44        if ((f(i) < f(i - step) && f(i) > f(i + step))) {
45          genAssert(f(i).abs() > 1.5, 'Точка слишком близко к 0');
46          funcDescendingPoints.push([i, f(i)]);
47        }
48    }
49    funcAscendingPoints = funcAscendingPoints.iz(2);
50    funcDescendingPoints = funcDescendingPoints.iz(2);
51
52    let points = [funcAscendingPoints.T()[0].concat(funcDescendingPoints.T()[0]),
53      funcAscendingPoints.T()[1].concat(
54        funcDescendingPoints.T()[1])].T();
55
56    let derivativeAsc = funcAscendingPoints.T()[0].map((x) => [x, (f(x) - f(x - 0.01)) *
57      100]);
```

```

55     let derivativeDesc = funcDescendingPoints.T()[0].map((x) => [x, (f(x) - f(x - 0.01))
56     * 100]);
57
58     let condition;
59     //TODO: придумать чтонибудь - получше
60     if ((derivativeAsc[0][1].abs() - derivativeAsc[1][1].abs()).abs() >= 1.5) {
61         condition = ['наибольшая'];
62         if (derivativeAsc[0][1] < derivativeAsc[1][1])
63             condition.push(derivativeAsc[1][0]);
64         else
65             condition.push(derivativeAsc[0][0]);
66     }
67     if ((derivativeDesc[0][1].abs() - derivativeDesc[1][1].abs()).abs() >= 1.5) {
68         condition = ['наименьшее'];
69         if (derivativeDesc[0][1] > derivativeDesc[1][1])
70             condition.push(derivativeDesc[1][0]);
71         else
72             condition.push(derivativeDesc[0][0]);
73     }
74     genAssert(condition.length != 1);
75
76     let pointsView = points.T()[0].map((num) => num.ts());
77
78     let paint1 = function(ct) {
79         let h = 380;
80         let w = 500;
81         ct.drawCoordinatePlane(w, h, {
82             hor: 1,
83             ver: 1
84         }, {
85             x1: '',
86             y1: '',
87             sh1: 17,
88         }, 20);
89
90         ct.fillStyle = "black";
91         ct.scale(20, -20);
92         ct.lineWidth = 0.15;
93
94         graph9AdrawFunction(ct, f, {
95             minX: minX,
96             maxX: maxX,
97             minY: -9,
98             maxY: 9,
99             step: 0.01
100         });
101
102         ct.fillStyle = "#0099ff";
103         graph9AmarkCircles(ct, points, points.length, 0.1);
104
105         ct.scale(1 / 20, -1 / 20);
106         ct.lineWidth = 2;
107         ct.fillStyle = "#007acc";
108         for (let i = 0; i < points.length; i++) {
109             ct.font = "17px liberation_sans";
110
111             if (f(points[i][0]) < 0) {
112                 ct.fillText(pointsView[i], points[i][0] * 20, -3);
113             } else {

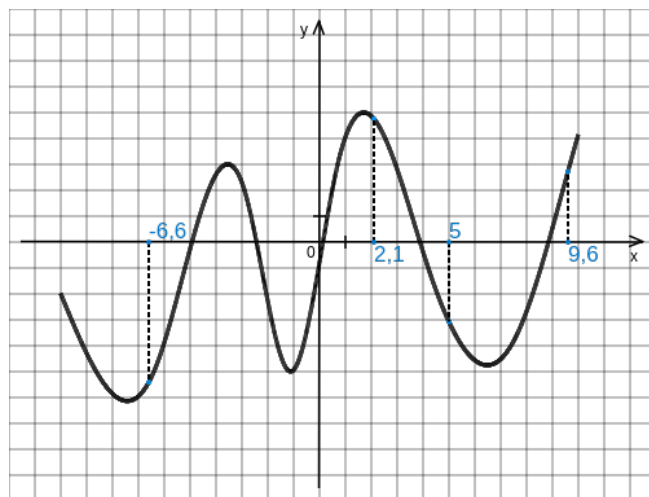
```

```

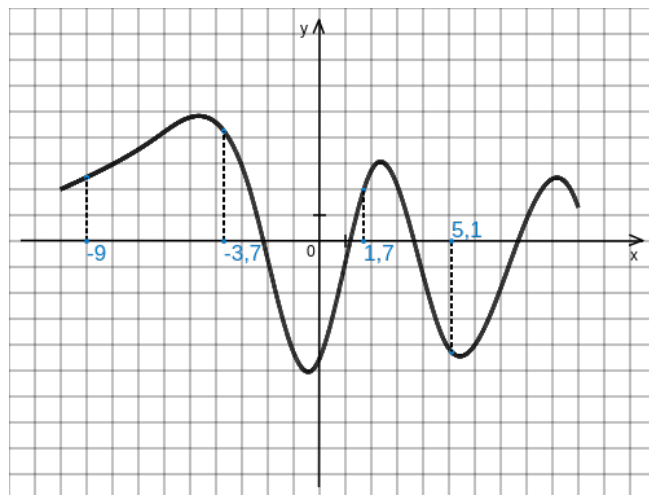
114         ct.fillText(pointsView[i], points[i][0] * 20, 15);
115     }
116
117     ct.setLineDash([4, 2]);
118     ct.drawLine(points[i][0] * 20, 0, points[i][0] * 20, -points[i][1] * 20);
119     graph9AmarkCircles(ct, [
120         [points[i][0] * 20, 0]
121     ], 1, 2);
122
123 }
124
125 };
126 NATask.setTask({
127     text: 'На рисунке изображен график функции  $y=f(x)$  и отмечены точки '$' +
pointsView.join('$; $') +
128     '$. В какой из этих точек значение производной ' + condition[0] + '? В ответе укажите
эту точку. ',
129     answers: condition[1],
130 });
131 chas2.task.modifiers.addCanvasIllustration({
132     width: 500,
133     height: 400,
134     paint: paint1,
135 });
136 });
137 })();
138 //SugarHedgehog
139 //317544 318045 318055 318139 318047 318049 318051 318053 318057 318059
140 //318061 318063 318065 318067 318069 318071 318073 318075 318077 318079
141 //318081 318083 318085 318087 318089 318091 318093 318095 318097 318099
142 //318121 318123 318125 318127 318129 318131 318133 318135 318137 318141 318143

```

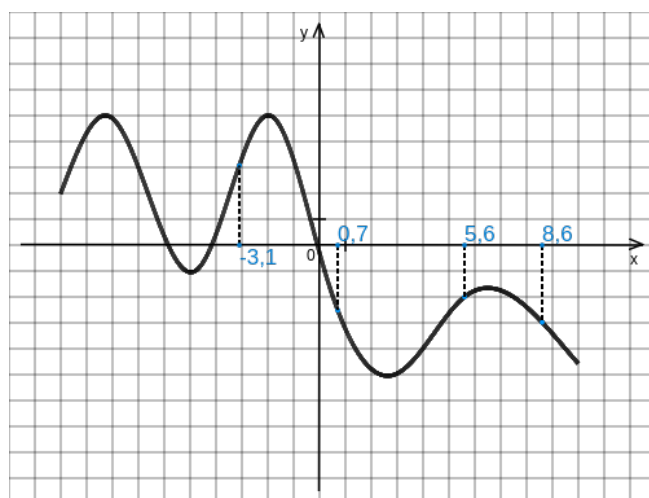
На рисунке изображен график функции $y = f(x)$ и отмечены точки $-6,6$; $9,6$; $2,1$; 5 . В какой из этих точек значение производной наибольшая? В ответе укажите эту точку.



На рисунке изображен график функции $y = f(x)$ и отмечены точки -9 ; $1,7$; $-3,7$; $5,1$. В какой из этих точек значение производной наибольшая? В ответе укажите эту точку.



На рисунке изображен график функции $y = f(x)$ и отмечены точки $5,6$; $-3,1$; $8,6$; $0,7$. В какой из этих точек значение производной наименьшее? В ответе укажите эту точку.



Заключение

В данной работе были приведены архитектура проекта «Час ЕГЭ», его библиотеки и примеры генерируемых задач. Обоснована релевантность проекта по сравнению с другими открытыми ресурсами.

В дальнейшем автор продолжит расширять каталог заданий по математике (в частности новых заданий номер 7 и 10) и будет способствовать появлению тестов по другим предметам.

Список литературы

- [1] Момот Е. А., Арахов Н. Д. Разработка и внедрение ПО для сбора статистики результатов подготовки к ЕГЭ по математике профильного уровня // Актуальные проблемы прикладной математики, информатики и механики. – 2021. – С. 1-2.
- [2] Открытый банк задач ЕГЭ по Математике. Профильный уровень. – URL: <https://prof.mathege.ru/>
- [3] Пошаговая инструкция по созданию элементарных шаблонов. – URL: <https://math.vsu.ru/chas-ege/doc/shabl-b1-po-shagam.html>
- [4] Федеральный институт педагогических измерений. – URL: <https://fipi.ru/ege/otkrytyy-bank-zadaniy-ege>
- [5] Единый государственный экзамен. – URL: https://ru.wikipedia.org/wiki/Единый_государственный_экзамен