МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра програмної інженерії та інформаційних технологій управління

Звіт з “Sprint 02”

з дисципліни «Чисельні методи»

Виконав:

ст. гр. КН-219б

Пономаренко В.С.

Перевірив:

проф. каф. ПІІТУ

Нікуліна О. М.

Харків

2021

ТЕМА: ЗНАХОДЖЕННЯ ЗНАЧЕНЬ ІНТЕГРАЛУ ТА РОЗВ’ЯЗУВАННЯ ДВ

ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

“

**Task 1.** Написать программу метода левых прямоугольников, программу оформить в виде функции. Программы можно писать на любом языке программирования. Входные параметры: f(x) – функция, интеграл которой находится, а, b – границы интегрирования, N – количество шагов. Выходные параметры: I – значение интеграла.

**Task 2.** Написать программу метода правых прямоугольников, программу оформить в виде функции. Программы можно писать на любом языке программирования. Входные параметры: f(x) – функция, интеграл которой находится, а, b – границы интегрирования, N – количество шагов. Выходные параметры: I – значение интеграла.

**Task 3.** Написать программу метода центральных прямоугольников, программу оформить в виде функции. Программы можно писать на любом языке программирования. Входные параметры: f(x) – функция, интеграл которой находится, а, b – границы интегрирования, N – количество шагов. Выходные параметры: I – значение интеграла.

**Task 4.** Написать программу метода трапеций, программу оформить в виде функции. Программы можно писать на любом языке программирования. Входные параметры: f(x) – функция, интеграл которой находится, а, b – границы интегрирования, N – количество шагов. Выходные параметры: I – значение интеграла.

**Task 5.** Написать программу метода парабол, программу оформить в виде функции. Программы можно писать на любом языке программирования. Входные параметры: f(x) – функция, интеграл которой находится, а, b – границы интегрирования, N – количество шагов. Выходные параметры: I – значение интеграла.

**Task 6.** Написать программу метода Эйлера, программу оформить в виде функции. Программы можно писать на любом языке программирования. Входные параметры: f(x, у) – функция, которая задана в правой части уравнения, находится, а, b – границы отрезка нахождения решения, y0 – начальное значение функции, N – количество шагов. Выходные параметры: У – массив значений функции на заданном отрезке.

**Task 7.** Написать программу метода Рунге-Кутта второго порядка, программу оформить в виде функции. Программы можно писать на любом языке программирования. Входные параметры: f(x, у) – функция, которая задана в правой части уравнения, находится, а, b – границы отрезка нахождения решения, y0 – начальное значение функции, N – количество шагов. Выходные параметры: У – массив значений функции на заданном отрезке.

**Task 8.** Написать программу метода Рунге-Кутта третьего порядка, программу оформить в виде функции. Программы можно писать на любом языке программирования. Входные параметры: f(x, у) – функция, которая задана в правой части уравнения, находится, а, b – границы отрезка нахождения решения, y0 – начальное значение функции, N – количество шагов. Выходные параметры: У – массив значений функции на заданном отрезке.

**Task 9.** Написать программу метода Рунге-Кутта четвертого порядка, программу оформить в виде функции. Программы можно писать на любом языке программирования. Входные параметры: f(x, у) – функция, которая задана в правой части уравнения, находится, а, b – границы отрезка нахождения решения, y0 – начальное значение функции, N – количество шагов. Выходные параметры: У –массив значений функции на заданном отрезке.

**Task 10.** Реализовать графический интерфейс пользователя программного продукта. Программный продукт предназначен для решения двух задач, поэтому при запуске программы, пользователь выбирает – вычисление определенного интеграла или решение дифференциального уравнения. Предусмотреть возможность выбора любого метода для решения выбранной задачи. Выбор целевых функций осуществляется из заданного списка. Cписок целевых функций

формирует сам разработчик программного продукта.

Например:

для вычисления интеграла:

1. f(x)=exp⁡(-x)

2. f(x)=sin⁡(x)

3. f(x)=exp⁡(-x^2)

4. f(x)=exp⁡(-4x-х^3) и т.д.

для решения ДУ:

1. ′ = −

2. ′= у + х

3. ′ = (3 − 12 2)

Предусмотреть возможность задания начальных условий, границ вычисления и количество шагов.

**Task 11**. Доработать и протестировать разработанный программный продукт. Для решения ДУ предусмотреть как табличный вывод результатов, так и графический. Предоставить сводную таблицу с результатами вычислений для определенного интеграла 5 методов, для одной из выбранной функции из списка. Проанализировать полученные результаты. Сделать выводы о работе методов.

”

МЕТА РОБОТИ

Метою виконання лабораторної роботи є написання програми для знахождення інтегралів та розв’язку ДУ.

1 ОПИСАННЯ РОЗРОБЛЕНОГО ЗАСТОСУНКУ

Програма була розроблена з використанням таких технологій:

- JavaScript;

- HTML;

- CSS;

Опис розробленого програмного застосунку я почну з “Task 10”, адже саме це завдання є підсумком усіх інших.

Для початку роботи з програмою вам потрібно, запустити Web сервер на вашому комп’ютері з кореневою папкою “t10”. Будь-які тестування рекомендую проводити в браузері “Google Chrome”.

Стартова сторінка проекту має такий вигляд, зображений на рисунку 1.

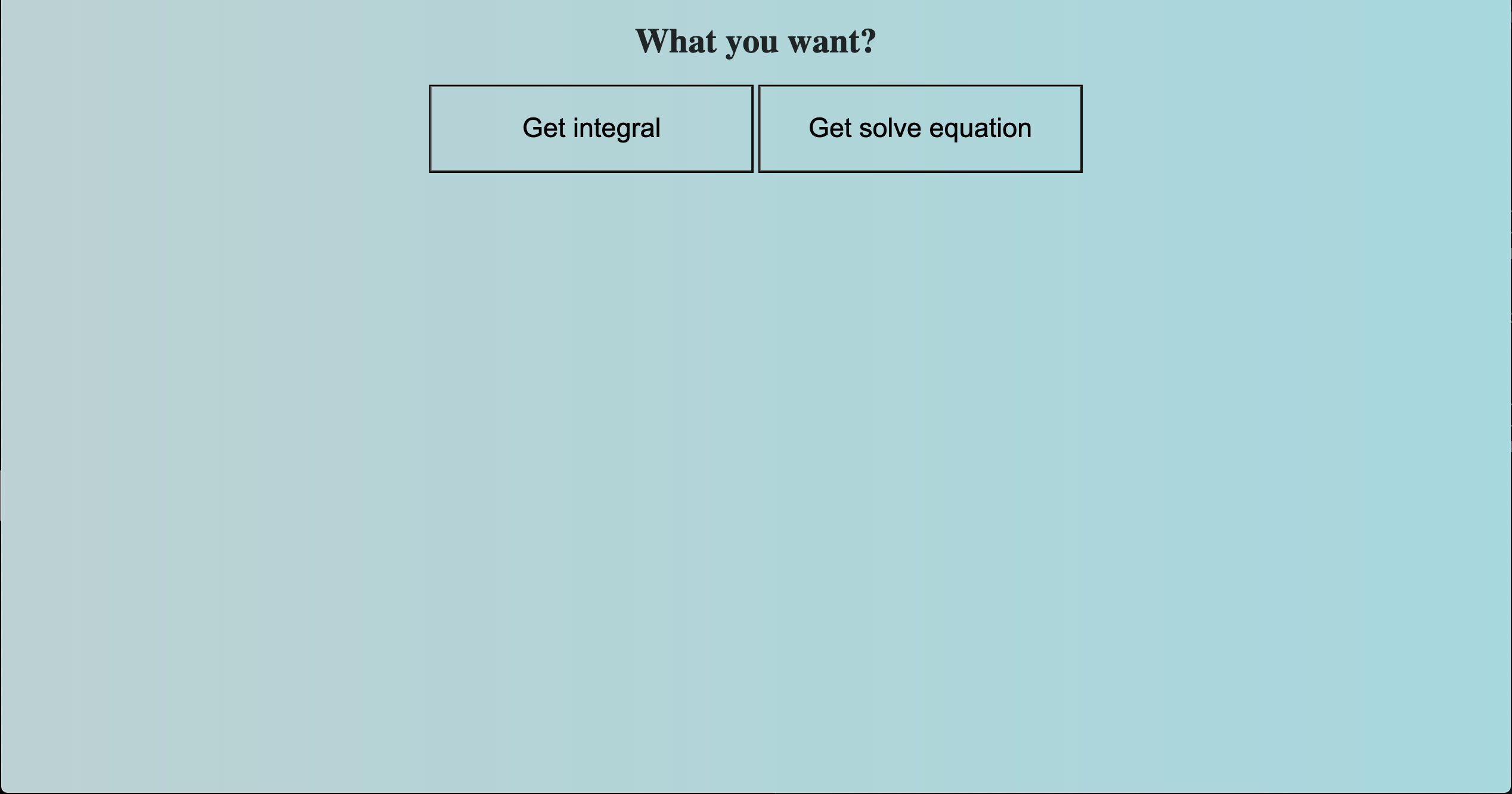
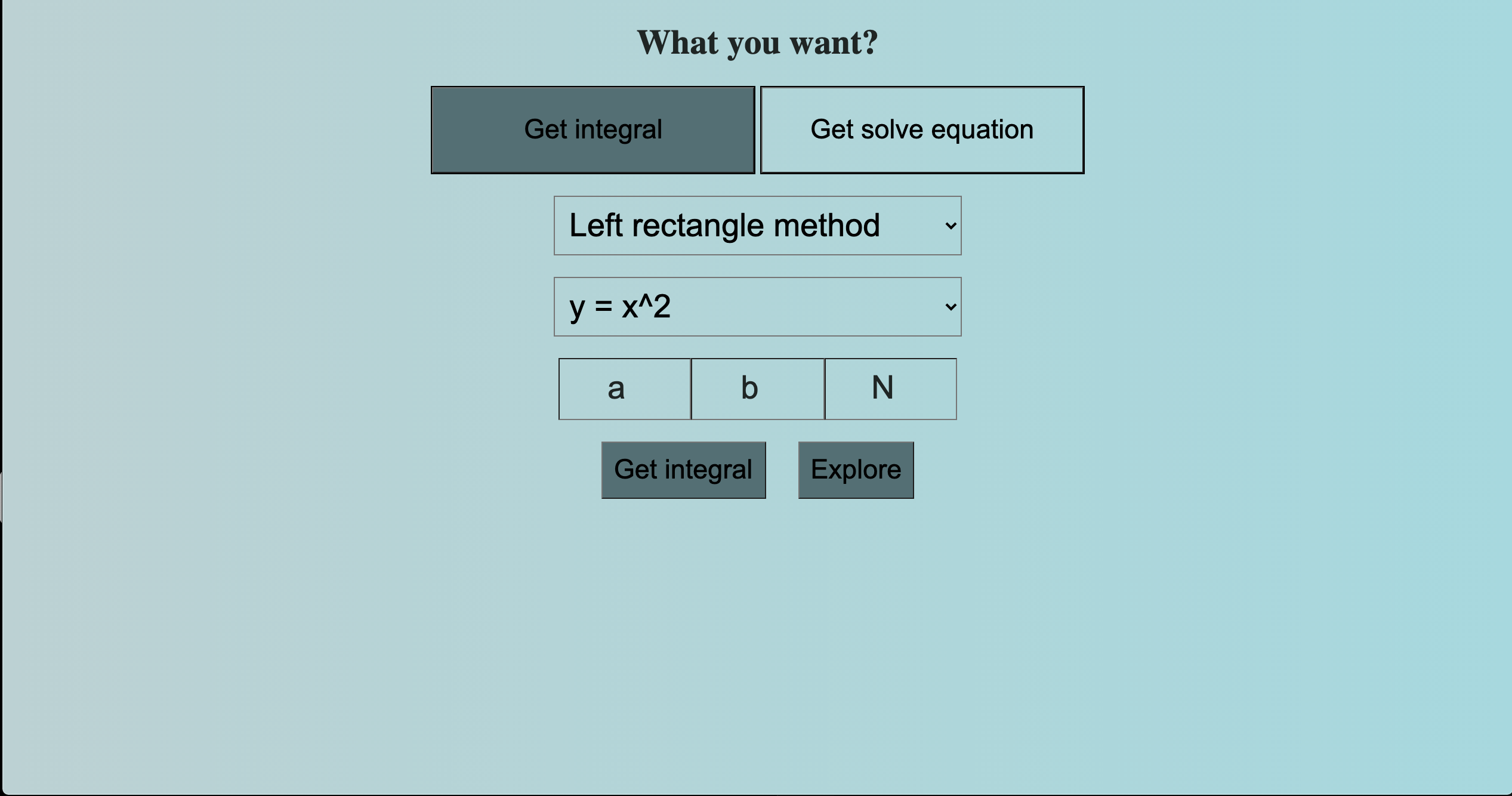
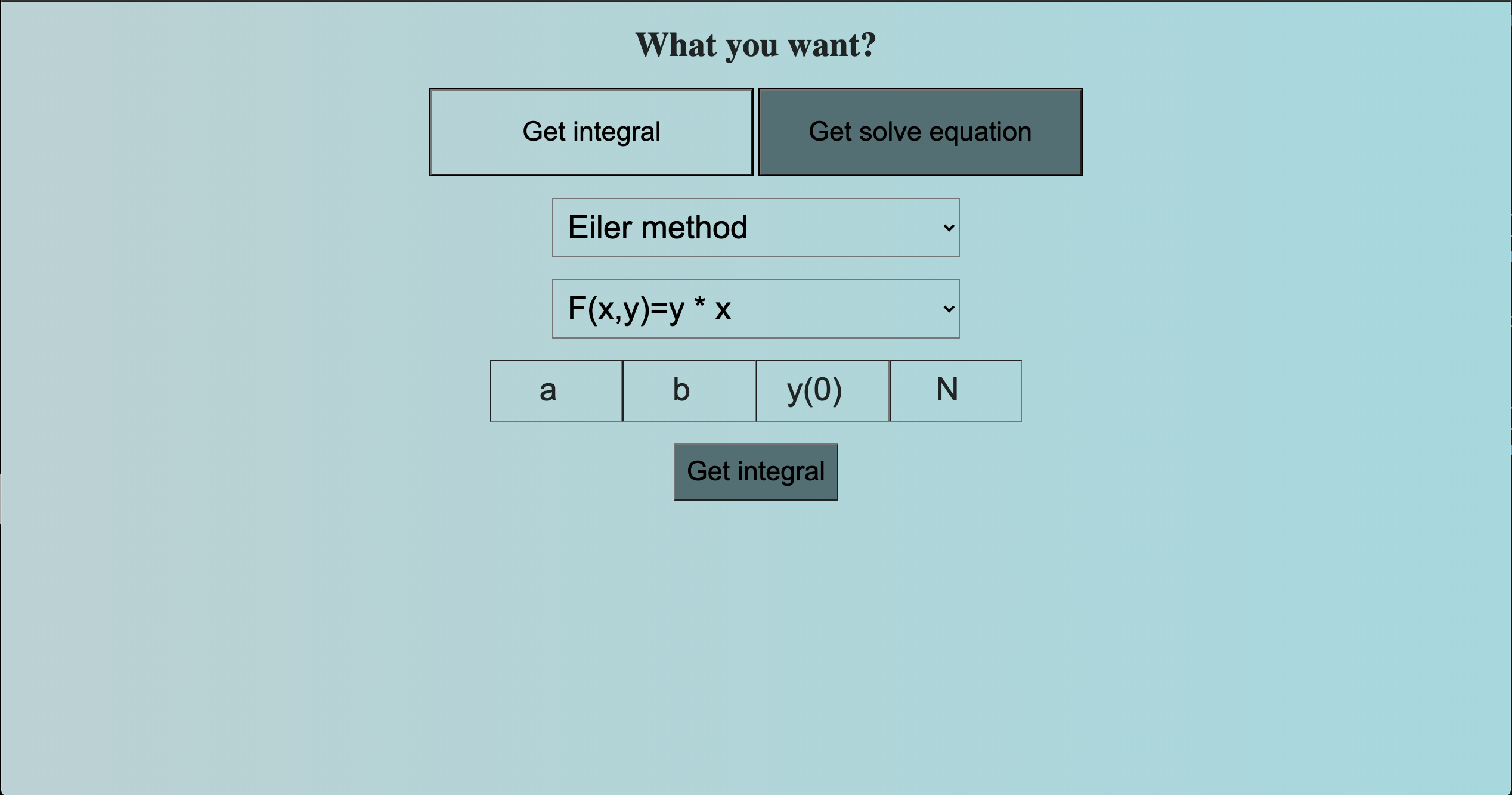


Рисунок 1 — Головна сторінка

Вам потрібно вибрати, що саме потрібно робити. Прогограма може знайти інтеграл, або розв’язати ДВ. (рис 2-3)

Рисунок 2 — Знайти інтеграл

Рисунок 3 — Розв’язати ДВ

Знаходження інтегралу та розв’язок ДВ, програмно розділені на дві одиниці трансляції, всю логіку пов’язану з інтегралом ви можете знайти в файлі «t10/js/integral\_main.js», а ДВ в файлі «t10/js/equation\_main.js», також в них є спільні допоміжні функції, які знаходяться в «t10/js/hp\_func.js», в цьому файлі також знаходятся функції для яких программа може проводити розрахунки.

«t10/js/integral\_main.js»:

*function print\_input\_menu\_differential\_equation() {*

*let integral\_main\_div = document.createElement("div");*

*integral\_main\_div.classList.add("way\_div");*

*integral\_main\_div.innerHTML = "" +*

*"<div class = 'user\_data\_div'>" +*

*"<select class = 'de\_select\_method'>" +*

*"<option value='1'>Eiler method</option>" +*

*"<option value='2'>Runge Kutta(2) method</option>" +*

*"<option value='3'>Runge Kutta(3) method</option>" +*

*"<option value='4'>Runge Kutta(4) method</option>" +*

*"</select>" +*

*"</div>" +*

*"<div class = 'user\_data\_div'>" +*

*"<select class = 'de\_select\_function'>" +*

*"<option value='1'>F(x,y)=y \* x</option>" +*

*"<option value='2'>F(x,y)=sin(x) \* y</option>" +*

*"<option value='3'>F(x,y)=x \* x + y \* y</option>" +*

*"</select>" +*

*"</div>" +*

*"<div class = 'user\_data\_div'>" +*

*"<input type='number' placeholder='a' id = 'de\_a\_value'>" +*

*"<input type='number' placeholder='b' id = 'de\_b\_value'>" +*

*"<input type='number' placeholder='y(0)' id = 'de\_y\_value'>" +*

*"<input type='number' placeholder='N' id = 'de\_N\_value'>" +*

*"</div>" +*

*"<div class = 'user\_data\_div'>" +*

*"<button id = 'get\_equation\_button'>Get integral</button>"*

*"</div>";*

*document.body.appendChild(integral\_main\_div);*

*}*

*function set\_equation\_button() {*

*let button = document.getElementById("get\_equation\_button");*

*button.onclick = () => {*

*let a = parseFloat(document.getElementById("de\_a\_value").value);*

*let b = parseFloat(document.getElementById("de\_b\_value").value);*

*let y0 = parseFloat(document.getElementById("de\_y\_value").value);*

*let N = parseInt(document.getElementById("de\_N\_value").value);*

*let func;*

*let func\_name;*

*switch (parseInt(document.getElementsByClassName("de\_select\_function")[0].value)) {*

*case 1:*

*func = de\_func1;*

*func\_name = "y` = x\*y";*

*break;*

*case 2:*

*func = de\_func2;*

*func\_name = "y` = x\*y";*

*break;*

*case 3:*

*func = de\_func3;*

*func\_name = "y` = x\*y";*

*break;*

*default:*

*func = de\_func4;*

*func\_name = "y` = x\*y";*

*break;*

*}*

*let method;*

*let name;*

*switch (parseInt(document.getElementsByClassName("de\_select\_method")[0].value)) {*

*case 1:*

*name = "Eiller - ";*

*method = d\_method\_eiler;*

*break;*

*case 2:*

*name = "Runge Kutta(2) - ";*

*method = d\_runge\_kutta\_2;*

*break;*

*case 3:*

*name = "Runge Kutta(3) - ";*

*method = d\_runge\_kutta\_3;*

*break;*

*case 4:*

*name = "Runge Kutta(4) - ";*

*method = d\_runge\_kutta\_4;*

*break;*

*default:*

*name = "Eiller - ";*

*method = d\_method\_eiler;*

*break;*

*}*

*////////////////////////////////////////////////////*

*console.log("A: "+a);*

*console.log("B: "+b);*

*console.log("Y: "+y0);*

*console.log("N: "+N);*

*console.log(method);*

*console.log(func);*

*////////////////////////////////////////////////////*

*delete\_result();*

*///console.log(method(func,a,b,y0,N));*

*let x\_arr = new Array(N);*

*let temp = a;*

*let h = (b-a)/N;*

*for (let i = 0; i < N; i++) {*

*x\_arr[i] = temp;*

*temp += h;*

*}*

*let y\_arr = method(func,a,b,y0,N);*

*fix\_arr(x\_arr);*

*fix\_arr(y\_arr);*

*print\_result(["X"],["Y"]);*

*print\_result(x\_arr,y\_arr);*

*print\_chart(x\_arr,y\_arr, func\_name);*

*}*

*}*

*function print\_chart(x\_arr,y\_arr, label) {*

*let chart\_div = document.createElement("div");*

*chart\_div.innerHTML = '<canvas id="speedChart" width="400" height="200"></canvas>';*

*chart\_div.classList.add("result\_div");*

*chart\_div.classList.add("result\_text");*

*document.body.appendChild(chart\_div);*

*var speedCanvas = document.getElementById("speedChart");*

*//Chart.defaults.global.defaultFontFamily = "Lato";*

*// Chart.defaults.global.defaultFontSize = 18;*

*var data\_chart = {*

*labels: x\_arr,*

*datasets: [{*

*label: label,*

*data: y\_arr,*

*lineTension: 0,*

*fill: false,*

*borderColor: 'black',*

*backgroundColor: "white"*

*},*

*]*

*};*

*var chartOptions = {*

*legend: {*

*display: true,*

*position: 'top',*

*}};*

*var lineChart = new Chart(speedCanvas, {*

*type: 'line',*

*data: data\_chart,*

*options: chartOptions*

*});}let button\_differential\_equation = document.getElementById("differential\_equation");*

*button\_differential\_equation.onclick = () => {*

*delete\_exist\_menu();*

*delete\_result();*

*button\_differential\_equation.style.backgroundColor = "rgb(76, 112, 117)";*

*print\_input\_menu\_differential\_equation();*

*set\_equation\_button();*

*}*

«t10/js/equation\_main.js»:

function print\_input\_menu\_integral() {

let integral\_main\_div = document.createElement("div");

integral\_main\_div.classList.add("way\_div");

integral\_main\_div.innerHTML = "" +

"<div class = 'user\_data\_div'>" +

"<select class = 'select\_method'>" +

"<option value='1'>Left rectangle method</option>" +

"<option value='2'>Right rectangle method</option>" +

"<option value='3'>Middle rectangle method</option>" +

"<option value='4'>Trapezoid method</option>" +

"<option value='4'>Parabol method</option>" +

"</select>" +

"</div>" +

"<div class = 'user\_data\_div'>" +

"<select class = 'select\_function'>" +

"<option value='1'>y = x^2</option>" +

"<option value='2'>y = sin(x)</option>" +

"<option value='3'>y = x^2 + 4 \* x + 25</option>" +

"</select>" +

"</div>" +

"<div class = 'user\_data\_div'>" +

"<input type='number' placeholder='a' required id = 'a\_value'>" +

"<input type='number' placeholder='b' required id = 'b\_value'>" +

"<input type='number' placeholder='N' required id = 'N\_value'>" +

"</div>" +

"<div class = 'user\_data\_div'>" +

"<button id = 'get\_integral\_button'>Get integral</button>" +

"<button id = 'explore\_integral\_button'>Explore</button>" +

"</div>";

document.body.appendChild(integral\_main\_div);

}

function set\_integral\_button() {

let button = document.getElementById("get\_integral\_button");

button.onclick = () => {

let a = parseFloat(document.getElementById("a\_value").value);

let b = parseFloat(document.getElementById("b\_value").value);

let N = parseInt(document.getElementById("N\_value").value);

let func;

switch (parseInt(document.getElementsByClassName("select\_function")[0].value)) {

case 1:

func = func1;

break;

case 2:

func = func2;

break;

case 3:

func = func3;

break;

default:

func = func1;

break;

}

let method;

let name;

switch (parseInt(document.getElementsByClassName("select\_method")[0].value)) {

case 1:

name = "Left Rectangle";

method = method\_left\_rectangle;

break;

case 2:

name = "Right Rectangle";

method = method\_right\_rectangle;

break;

case 3:

name = "Middle Rectangle";

method = method\_rectangle;

break;

case 4:

name = "Trapezoid";

method = method\_trapezoid;

break;

case 5:

name = "Parabol";

method = method\_parabol;

break;

default:

name = "Left Rectangle";

method = method\_left\_rectangle;

break;

}

////////////////////////////////////////////////////

console.log("A: "+a);

console.log("B: "+b);

console.log("N: "+N);

console.log(method);

console.log(func);

////////////////////////////////////////////////////

delete\_result();

let x\_arr = new Array(1);

x\_arr[0] = method(func,a,b,N).toFixed(5);

let name\_arr = new Array(1);

name\_arr[0] = name;

print\_result( name\_arr , x\_arr);

}

}

function set\_explore\_button() {

let button = document.getElementById("explore\_integral\_button");

button.onclick = () => {

let a = parseFloat(document.getElementById("a\_value").value);

let b = parseFloat(document.getElementById("b\_value").value);

let N = parseInt(document.getElementById("N\_value").value);

let func;

switch (parseInt(document.getElementsByClassName("select\_function")[0].value)) {

case 1:

func = func1;

break;

case 2:

func = func2;

break;

case 3:

func = func3;

break;

default:

func = func1;

break;

}

////////////////////////////////////////////////////

console.log("A: "+a);

console.log("B: "+b);

console.log("N: "+N);

console.log(func);

////////////////////////////////////////////////////

delete\_result();

let name\_arr = [

"Left Rectangle",

"Right Rectangle",

"Middle Rectangle",

"Parabol Rectangle",

"Trapezoid Rectangle"

];

let x\_arr = [

method\_left\_rectangle(func,a,b,N).toFixed(5),

method\_right\_rectangle(func,a,b,N).toFixed(5),

method\_rectangle(func,a,b,N).toFixed(5),

method\_parabol(func,a,b,N).toFixed(5),

method\_trapezoid(func,a,b,N).toFixed(5)

];

print\_result(name\_arr, x\_arr);

}

}

let button\_inegral = document.getElementById("integral");

button\_inegral.onclick = () => {

delete\_exist\_menu();

delete\_result();

button\_inegral.style.backgroundColor = "rgb(76, 112, 117)";

print\_input\_menu\_integral();

set\_integral\_button();

set\_explore\_button();

}

function fix\_arr(x\_arr) {

for (let i = 0; i < x\_arr.length; i++) {

x\_arr[i] = Number(x\_arr[i]).toFixed(3);

}

}

«t10/js/hp\_func.js»:

function delete\_exist\_menu() {

if(document.getElementsByClassName("way\_div")[0]) {

document.getElementsByClassName("way\_div")[0].remove();

document.getElementsByTagName("button")[0].style.backgroundColor = "transparent";

document.getElementsByTagName("button")[1].style.backgroundColor = "transparent";

}

}

function func1(x) {

return x\*x;

}

function func2(x) {

return Math.sin(x);}

function func3(x) {

return x\*x + 4\*x + 25;

}

function delete\_result() {

let arr = document.getElementsByClassName("result\_div");

let size = arr.length;

for (let index = 0; index < size; index++) {

arr[0].remove();

}

}

function de\_func1(x,y) {

return x\*y;

}

function de\_func2(x,y) {

return Math.sin(x)\*y;

}

function de\_func3(x,y) {

return x\*x+y\*y;

}

function print\_result(name\_arr,x\_arr) {

let res = document.createElement("div");

res.classList.add("result\_div");

res.classList.add("result\_text");

let str = "<table>";

for (let i = 0; i < name\_arr.length; i++) {

str += "<tr>";

str += "<td>";

str += name\_arr[i];

str += "</td>";

str += "<td>";

str += x\_arr[i];

str += "</td>";

str += "</tr>";

} str += "</table>";

res.innerHTML = str;

document.body.appendChild(res);

}

В завданнях 1-5, треба було реалізувати методи, зображені на рисунку 4.

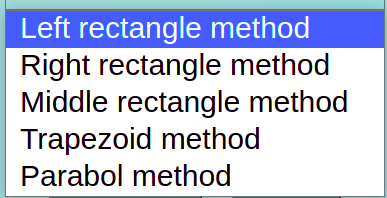


Рисунок 4 — Методи отримання інтегралу

1) Метод лівих прямокутників (t01):

function method\_left\_rectangle(func ,a, b, n) {

let result = 0.0;

let h = (b - a) / n;

for(let i = 0; i < n; i++) {

result += func(a)\*h;

a+=h;

}

return result;

}

2) Метод правих прямокутників (t02):

function method\_right\_rectangle(func ,a, b, n) {

let result = 0.0;

let h = (b - a) / n;

for(let i = 0; i < n; i++) {

result += func(b)\*h;

b-=h;

}

return result;

}

3) Метод центральних прямокутників (t03):

function method\_rectangle(func ,a, b, n) {

let result = 0.0;

let h = (b - a) / n;

for(let i = 0; i < n; i++) {

result += func(a+(h/2))\*h;

a+=h;

}

return result;

}

4) Метод трапецій:

function method\_trapezoid(func ,a, b, n) {

let result = 0.0;

let h = (b - a) / n;

for(let i = 0; i < n; i++) {

result += ((func(a) + func(a+h)) / 2)\*h;

a+=h;

}

return result;

}

5) Метод парабол:

function method\_parabol(func ,a, b, n) {

let h=(b-a)/n;

let k=0.0;

let x=a + h

for (let i = 1; i < n/2 + 1; i++) {

k += 4\*func(x);

x += 2\*h;

}

x = a + 2\*h;

for(let i = 1; i < n/2; i++) {

k += 2\*func(x)

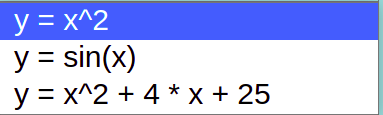
x += 2\*h

}

return (h/3)\*(func(a)+func(b)+k)

}

На рисунку 5, показоно функції для яких , програма може вести розрахунки.



Рисукок 5 — Доступні функції для інтегралу

У завданнях 6-9, потрібно було реазізувати методи для розв’язування ДВ.

1) Метод Эйлера:

function d\_method\_eiler (func, a,b,y0,N) {

let h = (b - a) / N;

let resut\_y = new Array(N);

resut\_y[0] = y0;

let x = a;

for (let index = 0; index < N; index++) {

resut\_y[index+1] = resut\_y[index] + h \* func(x,resut\_y[index]);

x += h;

}

return resut\_y;

}

1) Метод Рунги-Кутта(2):

function d\_runge\_kutta\_2 (func, a,b,y0,N) {

let h = (b-a)/N;

let res\_y = new Array(N);

res\_y[0] = y0;

let R1 = h\*func(a,res\_y[0]);

let R2 = h\*func(a+h/2, res\_y[0] + R1/2);

for (let i = 0; i < N; i++) {

res\_y[i+1] = res\_y[i] + (R1 + 2\*R2) / 6;

a += h;

R1 = h\*func(a,res\_y[i+1]);

R2 = h\*func(a+h/2, res\_y[i+1] + R1/2);

}

return res\_y;

}

1) Метод Рунги-Кутта(3):

function d\_runge\_kutta\_3 (func, a,b,y0,N) {

let h = (b-a)/N;

let res\_y = new Array(N);

res\_y[0] = y0;

let R1 = h\*func(a,res\_y[0]);

let R2 = h\*func(a+h/2, res\_y[0] + R1/2);

let R3 = h\*func(a+h/2, res\_y[0] + R2/2);

for (let i = 0; i < N; i++) {

res\_y[i+1] = res\_y[i] + (R1 + 2\*R2 + 2\*R3) / 6;

a += h;

R1 = h\*func(a,res\_y[i+1]);

R2 = h\*func(a+h/2, res\_y[i+1] + R1/2);

R3 = h\*func(a+h/2, res\_y[i+1] + R2/2);

}

return res\_y;

}

1) Метод Рунги-Кутта(4):

function d\_runge\_kutta\_4 (func, a,b,y0,N) {

let h = (b-a)/N;

let res\_y = new Array(N);

res\_y[0] = y0;

let R1 = h\*func(a,res\_y[0]);

let R2 = h\*func(a+h/2, res\_y[0] + R1/2);

let R3 = h\*func(a+h/2, res\_y[0] + R2/2);

let R4 = h\*func(a+h, res\_y[0] + R3);

for (let i = 0; i < N; i++) {

res\_y[i+1] = res\_y[i] + (R1 + 2\*R2 + 2\*R3 + R4) / 6;

a += h;

R1 = h\*func(a,res\_y[i+1]);

R2 = h\*func(a+h/2, res\_y[i+1] + R1/2);

R3 = h\*func(a+h/2, res\_y[i+1] + R2/2);

R4 = h\*func(a+h, res\_y[i+1] + R3);

}

return res\_y;

}

На рисунку 6, показоно функції для яких , програма може вести розрахунки.

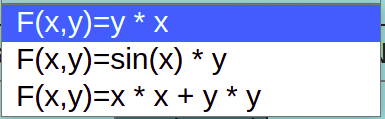


Рисунок 6 — Фукнкії для доступні для розв’язку ДВ

Інтерфейс взаємодії та його стиль були описані в файлах «t10/index.html» та «t07/css/style.css» відповідно.

«t07/index.html»:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Sprint02</title>

<link rel="stylesheet" href="css/style\_index.css">

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

</head>

<body>

<h1>What you want?</h1>

<div class="main\_select\_way">

<div class = "select\_way">

<button value="integral" id = "integral">Get integral</button>

</div>

<div class = "select\_way">

<button value="differential\_equation" id = "differential\_equation">Get solve equation</button>

</div>

</div>

<script src="js/integral/method\_left\_rectangle.js"></script>

<script src="js/integral/method\_right\_rectangle.js"></script>

<script src="js/integral/method\_rectangle.js"></script>

<script src="js/integral/method\_trapezoid.js"></script>

<script src="js/integral/method\_parabol.js"></script>

<script src="js/differential\_equation/d\_method\_eiler.js"></script>

<script src="js/differential\_equation/d\_method\_runge\_kutta\_2.js"></script>

<script src="js/differential\_equation/d\_method\_runge\_kutta\_3.js"></script>

<script src="js/differential\_equation/d\_method\_runge\_kutta\_4.js"></script>

<script src="js/hp\_func.js"></script>

<script src="js/integral\_main.js"></script>

<script src="js/equation\_main.js"></script>

</body>

</html>

«t07/css/style\_index.css» :

body{

color:rgb(30, 37, 36);

text-align: center;

background: linear-gradient(to left, rgb(153, 218, 223), rgb(183, 210, 212));

}

.select\_way {

text-align: center;

display: inline-block;

border: 1px black solid;

width: 300px;

height: 80px;

font-size: 25px;

}

.select\_way button {

background: none;

outline: none;

font-size: 25px;

width: 100%;

height: 100%;

}

select {

text-align: center;

width: 380px;

padding: 10px;

font-size: 30px;

background-color: rgba(153, 218, 223, 0);

}

.user\_data\_div {

margin: 20px;

}

.user\_data\_div input{

text-align: center;

width: 100px;

padding: 10px;

font-size: 30px;

background-color: rgba(153, 218, 223, 0);

}

.user\_data\_div input::placeholder {

color:rgb(30, 37, 36);

}

.user\_data\_div button {

font-size: 25px;

background-color: rgb(76, 112, 117);

padding: 10px;

outline: none;

margin-left: 15px;

margin-right: 15px;

}

.result\_text {

font-size: 40px;

}

table {

margin-left: 50%;

transform: translateX(-50%);

padding: 0;

}

td {

width: 150px;

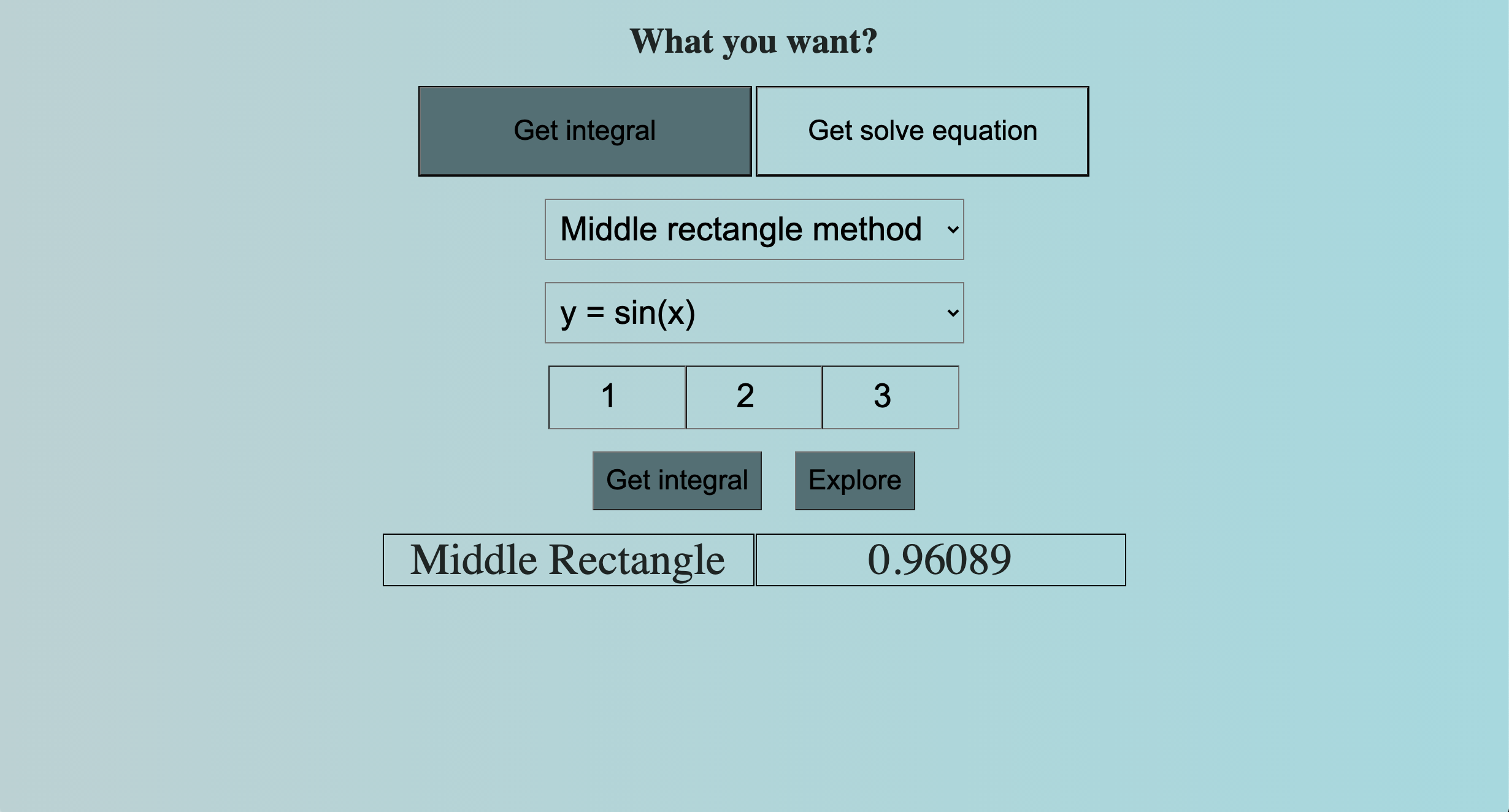
border: 1px black solid;

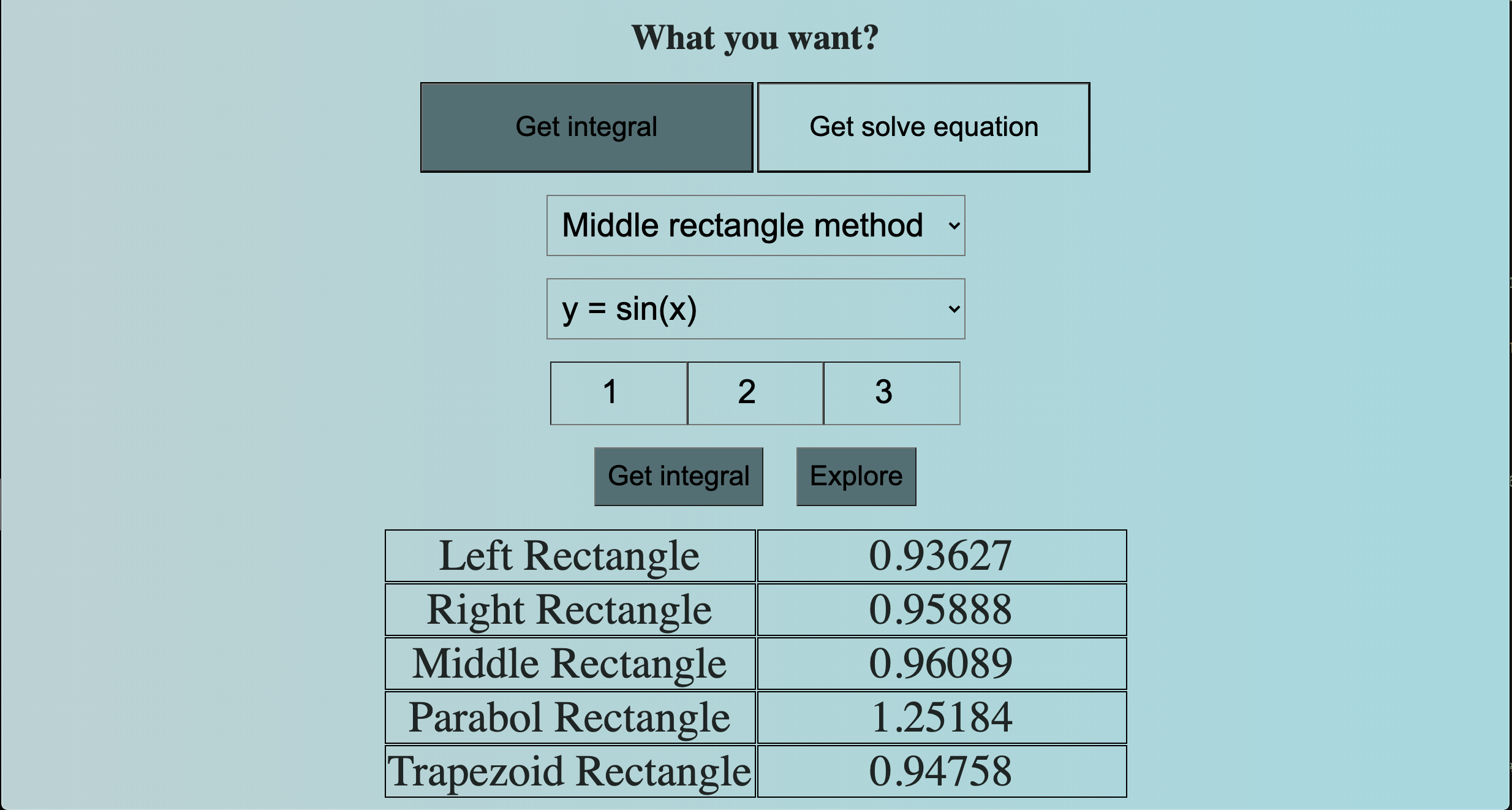
text-align: center;

}

2 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ЗАСТОСУНКУ

На рисунках 7-9 зображені тестування програмного застосунку.

Рисунок 7 — Тестування 1

Рисунок 8 — Тестування 2

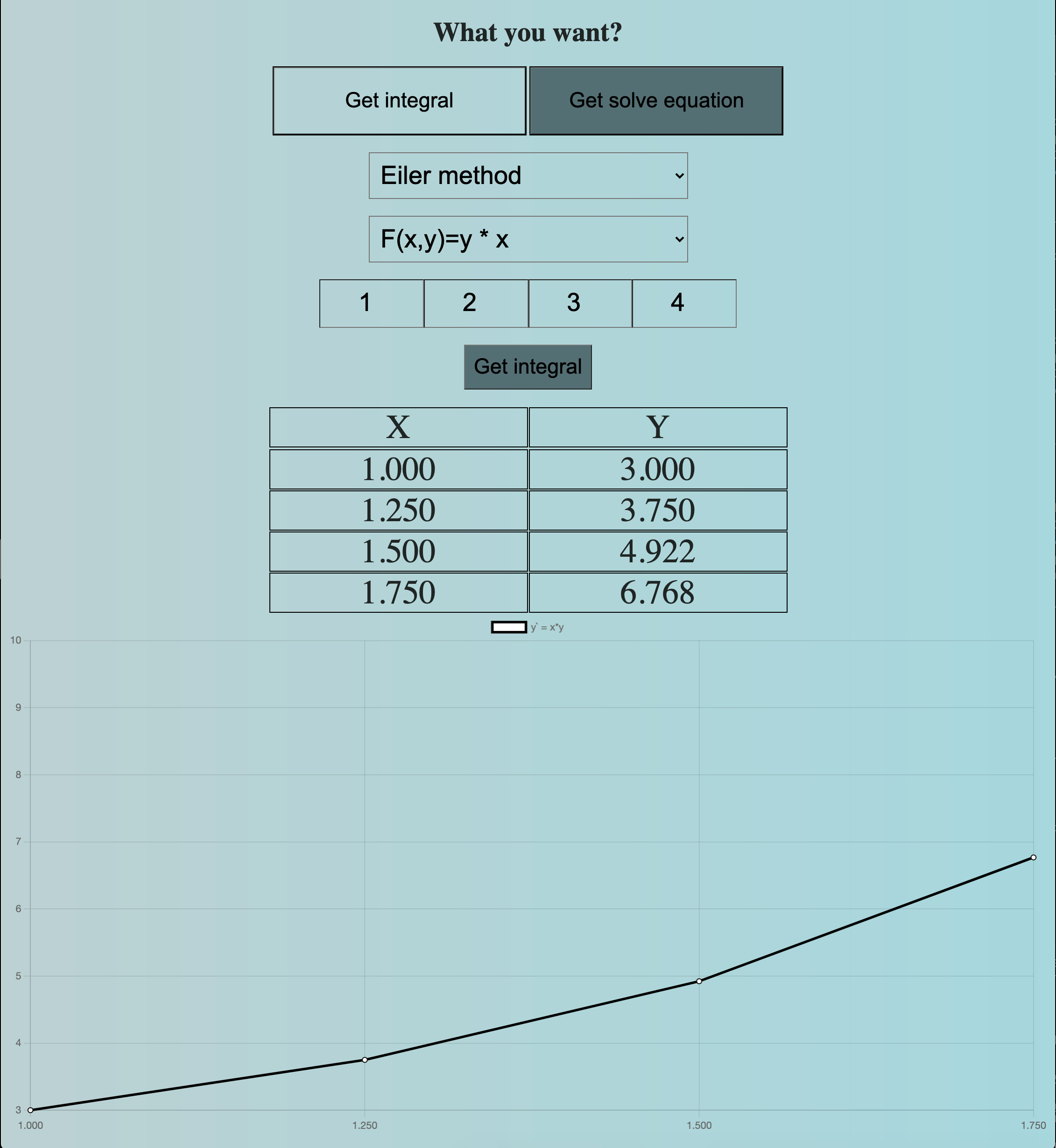


Рисунок 9 — Тестування 3

ВИСНОВКИ

Під час виконання даної лабораторної роботи була створена програма для знаходження інтегралів та розв’язування ДФ. Під час виконання цього спрінта, здобув навички при роботі з різними методами для знаходження інтегралів. З тяжкими трудостями не зіткнувся.