**INTERNET PROGRAMMING**Laboratory work 1-3  
Variant – 7

Prepared by:

student of *Ba-121-22-4-SE*,

*Anna Kuts`*

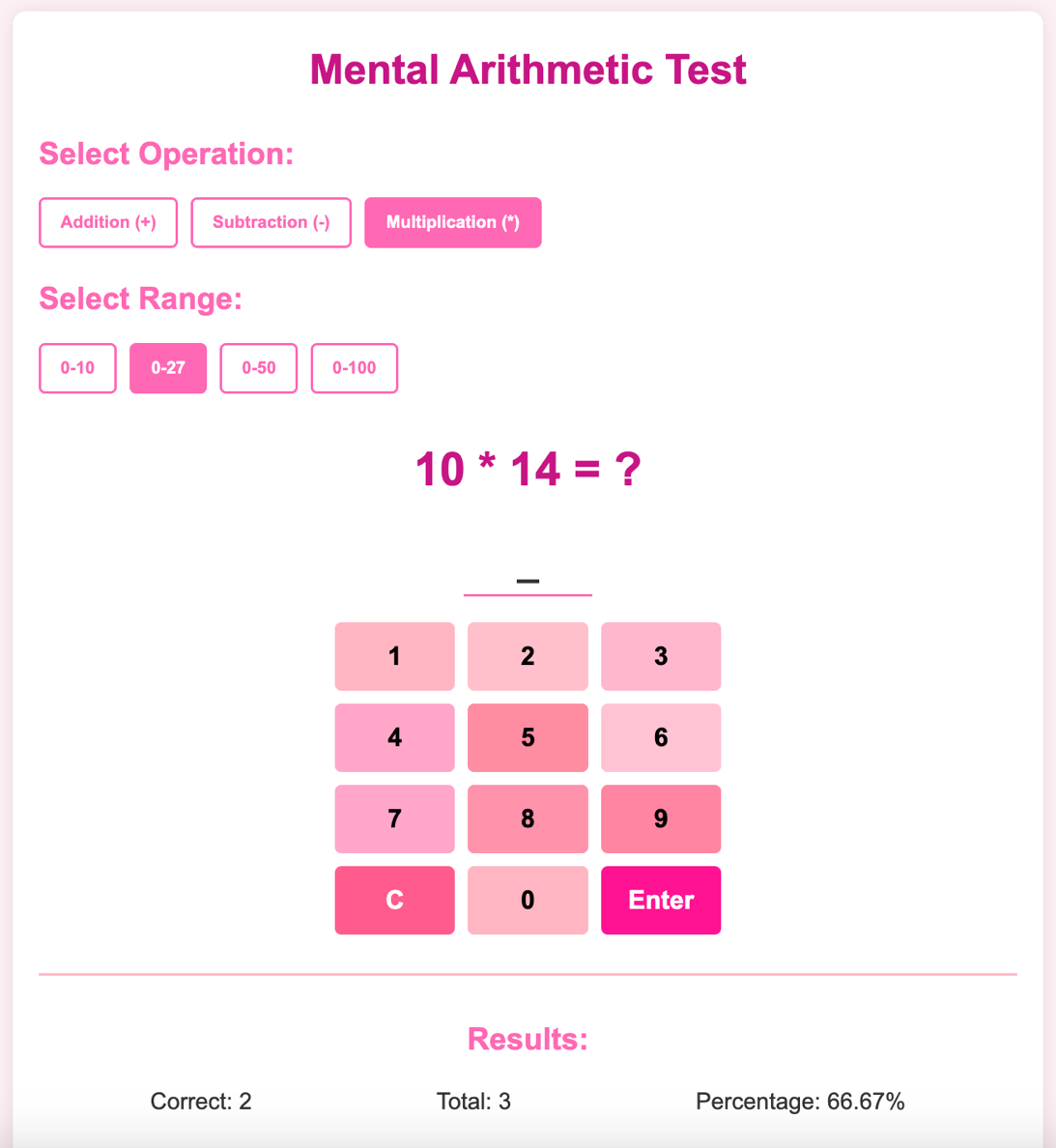
Accepted by:

*Makliuk O.O.*

**PART 1: Java Script   
Task**

Modify the script in test.html by adding a new range selection panel (as an option 20+student number in the group list), as well as the ability to check oral arithmetic for the arithmetic operation multiplication &quot;\*&quot;. Color the buttons of the number selection panel in different colors by adding a cascading style sheet file.

**Solution**

  
<https://github.com/Jasokaa/IP-Olenin>

**Conclusion**

In this work, I mastered the basic principles of JavaScript programming, understanded core concepts such as variables, data types, conditional statements, loops, and functions, and learnt how to create interactive web applications using this programming language.

1. What is JavaScript and what is it used for?  
JavaScript is a programming language used to create interactive and dynamic content on web pages (e.g., animations, form validation, event handling).

2. What data types exist in JavaScript? Give examples of each.

* String: "hello"
* Number: 42, 3.14
* Boolean: true, false
* Null: null
* Undefined: undefined
* Object: {name: "John", age: 30}
* Array: [1, 2, 3]
* Symbol: Symbol("id")
* BigInt: 1234567890123456789012345678901234567890n

3. How are variables declared in JavaScript? What is the difference between var, let, and const?

* var: function-scoped, can be redeclared.
* let: block-scoped, cannot be redeclared in the same scope.
* const: block-scoped, constant value, must be initialized.

4. What conditional statements are used in JavaScript? How do they work?

* if, else if, else — check conditions and execute blocks.
* switch — checks one expression against multiple cases.

5. What loops are available in JavaScript? Give examples of each.

* for: for (let i = 0; i < 5; i++) {}
* while: while (i < 5) {}
* do...while: do {} while (i < 5);
* for...of: for (let item of array) {}
* for...in: for (let key in object) {}

6. How are functions created in JavaScript? What is the difference between function declaration and function expression?

* Function declaration:

function greet() { console.log("Hello"); }

* Function expression:

const greet = function() { console.log("Hello"); };

* Declarations are hoisted, expressions are not.

7. How to handle events in JavaScript? Give examples of event handlers.  
Example using addEventListener:

document.getElementById("btn").addEventListener("click", function() {

alert("Button clicked!");

});

8. How to use arrays in JavaScript? Provide examples of adding, removing, and modifying array elements.  
In JavaScript, arrays are used to store multiple values in a single variable.

* Adding elements:

let arr = [1, 2, 3];

arr.push(4); // Adds 4 to the end

arr.unshift(0); // Adds 0 to the beginning

* Removing elements:

arr.pop(); // Removes last element

arr.shift(); // Removes first element

arr.splice(1, 1); // Removes one element at index 1

* Modifying elements:

arr[0] = 100; // Changes first element to 100

9. How to interact with the DOM (Document Object Model) in JavaScript? What methods are used to access and modify elements?  
JavaScript interacts with the DOM to dynamically access and change elements on a web page.

* Access elements:
  + document.getElementById("id")
  + document.querySelector(".class")
  + document.getElementsByTagName("div")
* Modify elements:

element.textContent = "New text";

element.innerHTML = "<b>Bold text</b>";

element.style.color = "blue";

element.classList.add("active");

10. What are the basic principles of working with asynchronous code in JavaScript? What constructs are used for asynchronous operations?  
Asynchronous code allows non-blocking execution, useful for tasks like API calls or timers.

* Main constructs:
  + Callbacks: Functions passed as arguments to be called later.
  + Promises: Handle asynchronous results using .then() and .catch().
  + async/await: Cleaner syntax to write asynchronous code.

Example with async/await:

async function getData() {

try {

let response = await fetch("https://api.example.com/data");

let data = await response.json();

console.log(data);

} catch (error) {

console.error("Error:", error);

}

}