**Лабораторна робота 8**

1. **Обмін**

Дано змінну a і змінну b (це означає: створіть змінні a і b та привласніть їм довільні значення, як у прикладі нижче). Змініть їх значення місцями. Тобто:

var a=10, b=-5; // …ваш код, який змінює змінні місцями

alert(a); // -5

alert(b); // 10

1. **Максимум**

Дано змінну a і змінну b (числа). Запишіть у змінну c максимальне значення з двох значень цих змінних. Не використовуйте модуль Math.

var a=10, b=-5; // …ваш код, який створює змінну “c” і записує в неї максимальне значення

alert(c); // 10

1. Дано змінну a. Запишіть у змінну b модуль числа a (тобто абсолютне значення, завжди невід’ємне). Не використовуйте модуль Math.

var a=-100; // …ваш код, який створює змінну “b” і записує в неї |a|

alert(b); // 100

1. **Формула**

Дано змінні x і z. Запишіть у змінну y значення, обчислене за формулою:

var x=-1, z=3.6; // …ваш код, який створює змінну “y” і записує в неї результат обчислення для будь-яких значень “x” і “z”

alert(y);

1. **Обмеження за віком**

Написати скрипт, який запитує вік людини (prompt), після чого просить підтвердити його (confirm) («Вам дійсно N років?»), після чого виводить повідомлення (alert) «Доступ дозволено», якщо людині є 18, або «Доступ заборонено» в іншому випадку.

У кінці вивести в консоль повідомлення про помилку (console.error), якщо в доступі було відмовлено, або повідомлення про успіх (console.log) в іншому випадку.

1. **Знак числа**

Попросити ввести будь-яке число.

Якщо воно опиниться більше за нуль, вивести повідомлення «Додатне число», якщо менше нуля – «Від’ємне число», якщо дорівнює нулю – «Дорівнює нуль!». Якщо введене взагалі не число, вивести повідомлення про помилку.

1. **Пароль**

Написати функцію, яка приймає у якості аргументів рядок і дві анонімні функції. Вона просить користувача ввести пароль. Якщо введено правильний пароль, що дорівнює введеному рядку, виконується перша функція, інакше – друга.

1. **Ділення націло**

Напишіть функцію canDivide(a, b), яка приймає два числа і повертає true, якщо a ділиться націло на b, або false – в іншому випадку.

alert(canDivide(9, 3)); // true

alert(canDivide(5, 2)); // false

alert(canDivide(10000, 5)); // true

alert(canDivide(101, 22)); // false

1. **Сума прогресії**

Напишіть функцію, яка приймає на вході число N і обчислює суму всіх чисел від 1 до N. Реалізуйте 3 варіанта функції (з 3 різними назвами) і порівняйте швидкість їхньої роботи на дуже великих числах:

- використовуючи рекурсію

- використовуючи цикл

- використовуючи формулу із шкільного курсу математики

1. **Фібоначчі**

Напишіть функцію, яка приймає параметр N і виводить у консоль N перших чисел із послідовності Фібоначчі. Реалізуйте два варіанти функції: - використовуючи рекурсію – використовуючи цикл

fibonacci(2); // 1, 1 (2 рядки в консолі)

Fibonacci(8); // 1, 1, 2, 3, 5, 8, 13, 21 (8 рядків у консолі)

1. **Безкінечне введення**

У циклі просити користувача ввести що-небудь доти, доки він не введе “exit”. Зберігати кожне введене значення в масив. У кінці – пройтися по отриманому масиву та вивести кожне значення за допомогою alert.

1. **Заміна в масиві**

Дано масив items. Напишіть функцію replace(x, y), яка приймає два параметри, проходить по масиву та заміняє всі елементи, що рівні x, на y.

var items = [3, 6, 0, 9, -1, 0, 0, 12, 0];

replace(0, 99);

alert(items); // 3, 6, 99, 9, -1, 99, 99, 12, 99

1. **Кількість парних**

Дано масив чисел numbers. Підрахуйте кількість парних елементів у ньому (тобто ваш код повинен підрахувати кількість парних елементів і вивести його на екран чи в консоль).

1. **Пошук максимуму**

Дано масив чисел numbers. Знайдіть максимальне число в масиві, використовуючи цикл.

1. **Середнє арифметичне**

Дано масив чисел numbers. Знайдіть середнє арифметичне значення чисел у масиві, використовуючи цикл.

1. **Від’ємні та додатні**

Дано масив чисел numbers. Підрахуйте кількість додатніх чисел, від’ємних чисел і нулів у масиві. Запишіть ці значення в 3 окремі змінні. Наприклад:

var items = [3, 6, 0, 9, -1, 0, 0, 12, 0];

var nPos = 0, nNeg = 0, nZero = 0; // …ваш код, який записує кількість додатних елементів у nPos, кількість від’ємних – у nNeg, кількість нулів – у nZero

alert(nPos); // 4

alert(nNeg); // 1

alert(nZero); // 4