

# Лабораторная работа №3. Использование сценариев на HTML страницах.

## Теоретическая часть

Web-документ, отображаемый браузером, – это результат исполнения программ, созданных на разных языках. Для описания структуры используется язык разметки (html), для описания внешнего вида – язык стилей (css). Для описания поведения документа, его реакции на действия пользователя используется язык сценариев (javascript).

JavaScript-код исполняет браузер. В него встроен интерпретатор JavaScript. Следовательно, выполнение программы зависит от того, когда этот интерпретатор получает управление. Опишем несколько способов размещения кода JavaScript на странице.

## Способы размещения кода JavaScript на странице

### 1. Использование тегового контейнера **<BODY>...</BODY>**

В теговом контейнере **<BODY> . . . </BODY>**.

```
<body>
  <script > команды скрипта</script>
</body>
```

### 2. Использование тегового контейнера **<HEAD>...</HEAD>**

В теговом контейнере **<HEAD> . . . </HEAD>** - если код скрипта представляет собой функцию, которая вызывается в ответ на какое-либо событие.

```
<head>
  <script type="text/javascript"> Здесь находятся команды сценария </script>
</head>
```

### 3. Использование внешнего файла

Во внешнем файле. По аналогии с тем, как стили подключаются к странице с помощью элемента `link`<sup>2</sup>, сценарии подключаются с помощью элемента `script`, только файл имеет расширение не `.css`, а `.js`.

```
<head>
<script type="text/javascript" src="my.js" > </script>
</head>
```

---

<sup>2</sup> HTML тег `<link>` определяет отношение между текущим html-документом и внешним ресурсом, на который он ссылается. Он является пустым элементом (не имеет содержимого и закрывающего тега) и всегда должен располагаться внутри элемента `<head>`. Чаще всего тег `<link>` используется для подключения внешних таблиц стилей:

```
<link href="style.css" rel="stylesheet" type="text/css">
```

Первый атрибут `href` определяет адрес документа, который может быть как абсолютным так и относительным. Второй атрибут `rel` указывает связь между HTML файлом и тем, на что вы ссылаетесь, в данном случае мы ссылаемся на таблицу стилей, поэтому используется значение `stylesheet`. Атрибут `type` сообщает браузеру MIME тип документа, для таблиц стилей значением всегда будет `"text/css"`.

## 4. Непосредственно в тегах

Обработчик события указывается прямо в теге, без заключения в теги `<script>`  
`</script>`

```
<input type="button" value="Нажать" onclick="window.alert('Нажмите еще раз');">
```

## Выполнение операторов сценария

Существует несколько способов определения момента запуска сценария. Вот некоторые из них:

- при загрузке документа;
- сразу после загрузки документа;
- в ответ на действия пользователя (используется чаще).

## Отладка скрипта

Если вы используете свой ноутбук, то скорее всего ваш браузер поддерживает встроенные средства отладки (Web developer tools), выясните способ их запуска. Например, в Mozilla Firefox они скрываются и отображаются с помощью сочетания клавиш «Ctrl+Shift+i». Если вы используете Internet Explorer, то нужно зайти в свойства обозревателя, вкладку «Дополнительно» и поставить галку в пункте «Показывать уведомления о каждой ошибке скрипта» в общем списке параметров.

## Синтаксис языка JavaScript

- JavaScript –зависит от регистра.
- Имена JavaScript и Javascript - разные имена!!
- Все ключевые слова используют только нижний регистр.
- Требования к именам переменных такие же, как в Си.
- Операторы разделяются точкой с запятой, которую можно опустить, если оператор заканчивается символом новой строки (Enter).

## Комментарии

```
// однострочный комментарий,  
/*  
..многострочный комментарий  
*/
```

## Типы данных

Переменные не имеют строгой типизации. Объявляются с помощью оператора `var`, который можно опускать, за исключением объявления локальных переменных в теле функции. Возможно объявление с одновременной инициализацией, например:

```
var s = 123 //объявляется целочисленная переменная x, имеющая значение 123  
var d=3.14 //объявляется переменная с плавающей точкой  
var str1='Строковая переменная'  
var p=true //объявляется логическая переменная
```

Тип переменной может изменяться в процессе выполнения программы. Если в выражении содержатся и числовые и строковые переменные, то числовые переменные автоматически приводятся к строковому типу.

## Математические операции

- + сложение
- - вычитание

- \*      умножение
- /      деление
- %      остаток от деления
- ++    увеличение значения на 1
- --    уменьшение значения на 1

### Операции сравнение

- ==    Равно
- !=    Не равно
- >    Больше
- <    Меньше
- >=   Больше либо равно
- <=   Меньше либо равно

### Логические операции

- &&    Логическая операции «И»
- ||    Логической операции «ИЛИ»
- !    Логическая операция «Отрицание»

### Операторы присваивания

- =      Присваивает значение переменной

также, как и в Си, имеют место сокращённые операции присвоения

- +=
- -=
- \*=
- /=
- %=

### Условные операторы и операторы циклов

По синтаксису их применение не отличаются от Си, пример использования цикла приводится в следующем в конце данной части примере.

Ниже пример ветвления:

```
if (x>=2 && x<=6) {
    y=0;
    z=1
} else {
    y=1; z=0
}
```

### Ввод/вывод данных. Диалоговые окна

В JavaScript существует 3 функции (метода), позволяющие пользователю выводить диалоговые окна:

- alert
- confirm
- prompt

#### **alert (“строка”)**

Метод alert используется для вывода простейшего диалогового окна, содержащего текст сообщения и единственную кнопку "Ok". Программа выводит сообщение и ожидает нажатия кнопки. После нажатия на кнопку, программа начинает выполняться дальше. Текст сообщения может сцепляться с любой текстовой переменной с помощью знака «+». Чтобы текст выводился в несколько строк используют символы «\n».

## **confirm (“строка”)**

Метод `confirm` используется в тех случаях, когда пользователь должен сделать выбор. Метод `confirm` позволяет пользователю вывести диалоговое окно, содержащее текст вопроса и кнопки "ОК" и "Отмена".

Функция `confirm` возвращает логическое значение в зависимости от нажатой пользователем кнопки:

- ОК соответствует значению `true`
- Отмена значению `false`

Результат работы функции может быть присвоен логической переменной, для дальнейшего анализа, как это показано в ниже следующем примере.

## **prompt (“строка1”, “строка2”)**

Метод `prompt` используется в тех случаях, когда пользователю нужно ввести значение в переменную. В окно выводится сообщение «строка1», в поле ввода помещается значение по умолчанию «строка2». Этот метод позволяет вывести диалоговое окно запроса на ввод данных. Результат работы функции присваивают переменной строкового типа.

## **Преобразование введённой строки к числовому типу**

Если введённые данные нужно использовать в арифметических выражениях, необходимо выполнить преобразование введённой строки к числовому типу. Это можно сделать при помощи следующих функций:

`parseInt("строка")` - преобразует строку в целое число;

`parseFloat("строка")` - преобразует строку в число с плавающей точкой.

## **Пример использования диалоговых окон и преобразования строки в число**

```
alert("Это alert.");
var result = confirm("Это второе диалоговое окно.\nВерно?");
if (result) {
    alert("Считать умеешь.");
} else {
    alert("Повтори азы арифметики, или пойдёшь выпить.");
}
var s = prompt("Введи любую строку", "А здесь уже что-то есть! Можно это удалить и написать своё!");
if (s != "") {
    alert("Всё же что-то ввёл?");
} else {
    alert("На нет и суда нет.");
}
var a = prompt("А можно ввести число, или даже много чисел. Введи первое...", "");
var b = prompt("Введи второе...", "");
alert("А это их сумма: " + (parseInt(a) + parseFloat(b)));
```

## **Метод document.write()**

В JavaScript есть также специальные встроенные объекты. Для упорядочивания огромного количества объектов существует объектная модель документа DOM. Эта модель является структурой организации объектов на странице. Объект `document` соответствует всему HTML-документу. Метод `write()` этого объекта позволяет динамически формировать документ.

`document.write("строка html-кода")` // выводит строку в окно документа

`document.writeln ("строка html-кода")` /\* выводит строку в окно документа, в конце выводится символ "пробел" \*/

### Пример №1. Использование метода `document.write()`

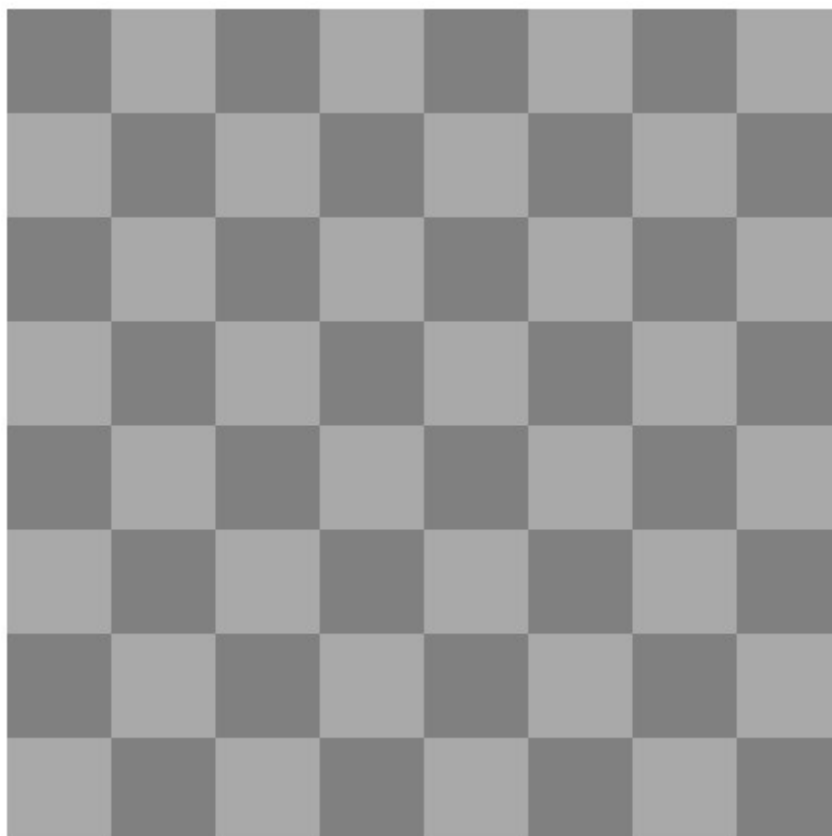
Метод, применяемый к объекту, пишется после имени объекта через точку. Содержимое строки должно быть в кавычках или это может быть объединение (сумма) нескольких строк или строковых переменных. Строка должна содержать элементы разметки страницы (теги и их содержимое). Метод выполняется в процессе загрузки документа.

```
var i, n = 3;
for (i = 0; i < n; i++) {
    document.write("<p>Парграф №" + (i + 1) + "</p>");
}
```

### Пример №2. Использование метода `document.write()`

Рассмотрим ещё один пример. Создадим таблицу 8 на 8 (шахматная доска). Высота и ширина ячейки 50 пикселей. Закрасим таблицу в шахматном порядке. Таблицу будем формировать динамически, с помощью метода `write`. Для закраски таблицы используем вложенные операторы цикла.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title></title>
</head>
<style type="text/css">
    table {
        border-collapse: collapse;
    }
    td {
        width: 50px;
        height: 50px;
    }
    .r1 {
        background-color: gray;
    }
    .r2 {
        background-color: darkgray;
    }
</style>
<body>
<script type="text/javascript">
var str, sel, n = 8, i, j;
document.write("<table>");
for (i = 0; i < n; i++) {
    document.write("<tr>");
    for (j = 0; j < n; j++) {
        sel = (i + j)%2;
        if (sel == 0) {
            str = "class='r1'";
        } else {
            str = "class='r2'";
        }
        document.write("<td " + str + "></td>");
    }
    document.write("</tr>");
}
document.write("</table>");
</script>
</body>
</html>
```



**Рис. 3.1 Таблица 8 на 8.**

До цикла и после цикла выводятся теги `<table>` и `</table>`. Во внешнем цикле в документ выводятся теги `<tr>` и `</tr>`, формирующие строку таблицы. Во внутреннем цикле в документ выводятся теги `<td>` и `</td>`, формирующие ячейки таблицы. От чётности `sel` зависит значение текстовой переменной `str`, которая определяет класс ячейки. Ячейка класса `r1` имеет серую заливку, ячейка класса `r2` заливается тёмносерым цветом.

### **Пример №3. Использование метода `document.write()`**

Рассмотрим ещё один пример. Создадим таблицу 8 на 8 (шахматная доска). Высота и ширина ячейки 100 пикселей. Закрасим таблицу в шахматном порядке. Таблицу будем формировать динамически, с помощью метода `write`. Для закрашки таблицы используем вложенные операторы цикла. В левом верхнем углу разместим чёрную шашку, а в нижнем правом углу разместим белую шашку.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title></title>
</head>
<style type="text/css">
  table {
    border-collapse: collapse;
    border: 2px solid;
  }
  td {
    width: 100px;
    height: 100px;
    border: 2px solid;
  }
  .r1 {
    background-color: gray;
```

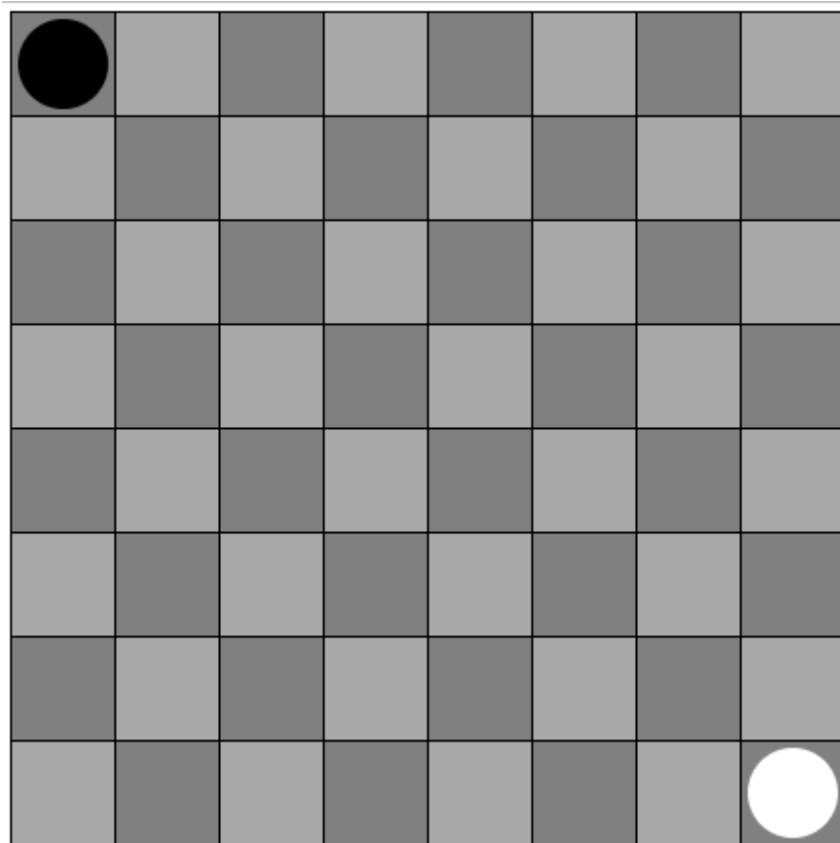
```

    }
    .r2 {
        background-color: darkgray;
    }
    .circle_black {
        background: black;
        border-radius: 50%;
        height: 90px;
        width: 90px;
        margin: 0px 0px 0px 5px;
    }

    .circle_white {
        background: white;
        border-radius: 50%;
        height: 90px;
        width: 90px;
        margin: 0px 0px 0px 5px;
    }
}

</style>
<body>
<script type="text/javascript">
var str, sel, n = 8, i, j;
document.write("<table>");
for (i = 0; i < n; i++) {
    document.write("<tr>");
    for (j = 0; j < n; j++) {
        sel = (i + j)%2;
        if (sel == 0) {
            str = "class='r1'";
        } else {
            str = "class='r2'";
        }
        if(i == 0 && j == 0){
            document.write("<td " + str + "><div
class='circle_black'></div></td>");
        }else if(i == 7 && j == 7){
            document.write("<td " + str + "><div
class='circle_white'></div></td>");
        }else{
            document.write("<td " + str + "></td>");
        }
    }
    document.write("</tr>");
}
document.write("</table>");
</script>
</body>
</html>

```



**Рис. 3.1 Таблица 8 на 8 и две шашки.**

## Рекомендации

Используйте удобный вам текстовый редактор для создания HTML страниц и сценариев на javascript.

Далее приводится пример HTML файла, который вы можете использовать как основу для разработки, отладки и показа вашего сценария на javascript, выполняемого в соответствии с вашим индивидуальным заданием.

## Примерная HTML страницы

```
<!DOCTYPE html>

<html>
<head>
  <meta charset="UTF-8"><title>Пример оформления л/работы по javascript</title>
</head>

<body>

  <table border="solid">
    <tr>
      <td>Параметр №1:</td>
      <td>
        <input type="text" id="parametr_1">
        <input type="button" value="Вычислить №1" onclick="calculate_1();">
      </td>
    </tr>
    <tr>
      <td>Результат №1:</td>
      <td><p id="result_1"></p>
    </tr>
  </table>
```



```

        <td>Параметр №2:</td>
        <td>
            <input type="text" id="parametr_2">
            <input type="button" value="Вычислить №2" onclick="calculate_2();">
        </td>
    </tr>
    <tr>
        <td>Результат №2:</td><td><p id="result_2"></td>
    </tr>
</table>

<script type="text/javascript">
    document.write("<br>");
    var i;
    for(i = 1; i < 10; i++) {
        document.write(i + ",");
    }
    document.write(i);
    document.write("<br>");
</script>

</body>

<script type="text/javascript">

function calculate_1() {
    var parametr_1 = document.getElementById("parametr_1").value;
    var result_1 = document.getElementById("result_1");
    result_1.innerHTML = Number(parametr_1) + 1234;
    //result_1.innerHTML = parseInt(parametr_1) + 1234;
    //result_1.innerHTML = (parametr_1 - 0) + 1234;
}

function calculate_2() {
    var parametr_2 = document.getElementById("parametr_2").value;
    var result_2 = document.getElementById("result_2");
    result_2.innerHTML = Math.exp(parseFloat(parametr_2));
}

</script>

</html>

```

Приведённая в примере HTML — страница выглядит в браузере примерно следующим образом, рисунок 3.2:

Параметр №1:	<input type="text" value="12"/>	Вычислить №1
Результат №1:	<input type="text"/>	
Параметр №2:	<input type="text" value="22"/>	Вычислить №2
Результат №2:	<input type="text"/>	

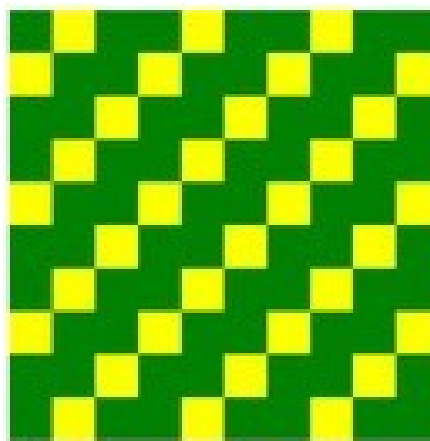
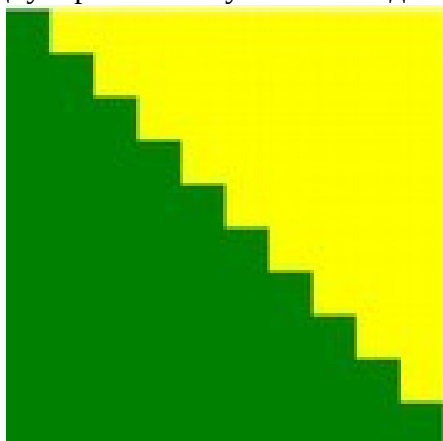
1,2,3,4,5,6,7,8,9,10

Рис. 3.2.

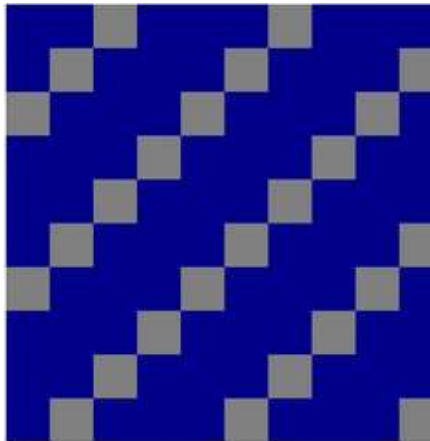
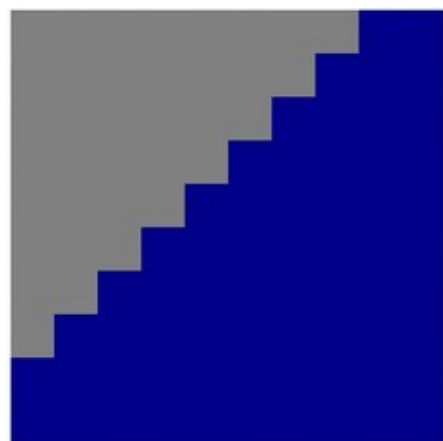
## Практическая часть

1. Ознакомится с теоретической частью.
2. Показать работу html — страницы, соответствующий индивидуальному варианту задания.
3. С помощью функций ввод/вывод данных и диалоговых окон необходимо выполнить одну из следующих последовательностей действий:
  - 1) — Запросить ввести свои фамилию, имя и отчество. Запросить подтверждение корректности ввода (показав введённые ФИО), если ввод был корректный вывести приветствие, в противном случае сообщение об ошибке.
  - 2) — Запросить два числа  $x$ ,  $y$  и знак арифметической операции (+, −, \*, /). Найти соответственно их сумму  $x+y$ , или разность  $x-y$ , или произведение  $x*y$ , или частное  $x/y$ . В случае ошибки в знаке арифметической операции или деления на 0 вывести сообщение об ошибке.
  - 3) — Запросите ввод времени в часах и минутах. Определите время, которое будет через минуту. Предусмотреть три случая: а) увеличение только минут; б) увеличение числа часов обнуление числа минут; в) переход через полночь (00:00)
4. Вывести на html-страницу либо один из следующих рисунков либо одну из определённых надписей, построенную из раскрашенных ячеек таблицы, либо исходную расстановку шашек на доску 8 на 8 клеток.

1) —



2) —



- 3) — По аналогии и в соответствии с примером из раздела «Пример №2. Использование метода `document.write()`» из закрашенных ячеек таблицы построить надпись «КАИ».
- 4) — По аналогии и в соответствии с примером из раздела «Пример №2. Использование метода `document.write()`» из закрашенных ячеек таблицы построить надпись «КНИТУ-КАИ».
- 5) — По аналогии и в соответствии с примером из раздела «Пример №2.

Использование метода `document.write()` из закрашенных ячеек таблицы построить надпись «ИКТЗИ».

6) — По аналогии и в соответствии с примером из раздела «Пример №2.

Использование метода `document.write()` из закрашенных ячеек таблицы построить надпись «ПМИ».

7) — По аналогии и в соответствии с примером из раздела «Пример №3.

Использование метода `document.write()` из закрашенных ячеек таблицы построить расстановку шашек на доску 8 на 8. Чёрные слева.

8) — По аналогии и в соответствии с примером из раздела «Пример №3.

Использование метода `document.write()` из закрашенных ячеек таблицы построить расстановку шашек на доску 8 на 8. Чёрные справа.

9) — По аналогии и в соответствии с примером из раздела «Пример №3.

Использование метода `document.write()` из закрашенных ячеек таблицы построить расстановку шашек на доску 8 на 8. Чёрные сверху.

10) — По аналогии и в соответствии с примером из раздела «Пример №3.

Использование метода `document.write()` из закрашенных ячеек таблицы построить расстановку шашек на доску 8 на 8. Чёрные снизу.

5. С учётом рекомендаций приведённых в разделе «Рекомендации» продемонстрировать решение двух задач из раздела «Список задач».

### Список задач

1. Дано натуральное число  $n$ . Переставить местами первую и последнюю цифры этого числа.
2. Даны два натуральных числа  $m$  и  $n$  ( $m < 9999$ ,  $n < 9999$ ). Проверить, есть ли в записи числа  $m$  цифры, совпадающие с цифрами в записи числа  $n$ .
3. Дано натуральное число  $n$ . Проверить, есть ли в записи числа три одинаковых цифры ( $n < 9999$ ).
4. Дано натуральное число. Дописать к нему цифру  $k$  в конец и в начало.
5. Даны натуральные числа  $n$ ,  $k$ . Проверить, есть ли в записи числа  $n^k$  цифра  $m$ .
6. Среди всех  $n$ -значных чисел указать те, сумма цифр которых равна данному числу  $k$ .
7. Заданы три натуральных числа  $A$ ,  $B$ ,  $C$ , которые обозначают число, месяц и год. Найти порядковый номер даты, начиная отсчет с начала года.
8. Найти наибольшую и наименьшую цифры в записи данного натурального числа.
9. Дано натуральное число  $N$ . Найти и вывести все числа в интервале от 1 до  $N - 1$ , у которых сумма всех цифр совпадает с суммой цифр данного числа. Если таких чисел нет, то вывести слово «нет».
10. Дано натуральное число  $N$ . Найти и вывести все числа в интервале от 1 до  $N - 1$ , у которых произведение всех цифр совпадает с суммой цифр данного числа. Если таких чисел нет, то вывести слово «нет».
11. Дано натуральное число  $N$ . Определить количество 8-значных чисел, у которых сумма цифр в цифровой записи числа меньше, чем  $N$ . Если таких чисел нет, то вывести слово «нет».
12. Дано натуральное число  $N$ . Определить количество 8-значных чисел, у которых сумма цифр в цифровой записи числа больше, чем  $N$ . Если таких чисел нет, то вывести слово «нет».
13. Дано натуральное число  $N$ . Найти наибольшее число  $M$  ( $M > 1$ ), на которое сумма цифр в цифровой записи числа  $N$  делится без остатка. Если такого числа нет, то вывести слово «нет».
14. Дано натуральное число  $N$ . Найти наименьшее число  $M$  ( $N < M < 2N$ ), которое делится на сумму цифр числа  $N$  (без остатка). Если такого числа нет, то вывести слово «нет».
15. Дано натуральное число  $N$  ( $N > 9$ ). Определить количество нулей, идущих подряд в младших разрядах данного числа.

16. Дано натуральное число  $N$  ( $N > 9$ ). Определить количество нулей в цифровой записи числа, кроме нулей в младших разрядах.
17. Дано натуральное число  $N$  ( $N > 9$ ). Определить сумму цифр в первой половине числа (старшие разряды).
18. Дано натуральное число  $N$  ( $N > 9$ ). Определить сумму цифр во второй половине числа (младшие разряды).
19. Женщина шла на базар продавать яйца. Ее случайно сбил с ног всадник, в результате чего все яйца разбились. Всадник предложил оплатить убытки и спросил, сколько у нее было яиц. Женщина сказала, что точного числа не помнит, но когда она брала яйца парами, то оставалось одно яйцо. Одно яйцо оставалось также, когда она брала по 3, 4, 5 и 6 яиц, но когда она брала по 7 штук, то в остатке ничего не было. Какое минимальное число яиц могло быть в корзине?
20. Дано натуральное число  $N$ . Поменять порядок следования цифр в этом числе на обратный или сообщить, что это невозможно в силу переполнения.
21. Дано натуральное число  $N$ . Определить количество цифр в цифровой записи данного числа, которые имеют наименьшее значение.
22. Дано натуральное число  $N$ . Определить количество цифр в цифровой записи данного числа, которые имеют наибольшее значение.
23. Дано натуральное число  $N$ . Получить новое число  $M$ , которое образуется из числа  $N$  путем замены последней цифры на значение наименьшей цифры в записи числа  $N$ .
24. Дано натуральное число  $N$ . Получить новое число  $M$ , которое образуется из числа  $N$  путем замены последней цифры на значение наибольшей цифры в записи числа  $N$ .
25. Определить количество  $M$ -значных натуральных чисел, у которых сумма цифр, стоящих в нечетных разрядах, равна  $N$  ( $1 < N < 90$ ,  $5 < M < 20$ ).
26. Вычислить сумму тех чисел из заданного отрезка  $[a, b]$  ( $a, b$  — натуральные), в запись которых входит цифра  $k$ .
27. Дано натуральное  $k$ . Напечатать  $k$ -ю цифру последовательности 12345678910111213..., в которой выписаны подряд все натуральные числа.
28. Дано натуральное  $k$ . Напечатать  $k$ -ю цифру последовательности 149162536..., в которой выписаны подряд квадраты всех натуральных чисел.
29. Дано натуральное число  $N$ . Переставить его цифры так, чтобы образовалось максимальное число, записанное теми же цифрами.
30. Дано натуральное число  $N$ . Переставить его цифры так, чтобы образовалось наименьшее число, записанное теми же Цифрами.
31. Дано натуральное  $k$ . Напечатать  $k$ -ю цифру последовательности 24681012141618202224262830..., в которой выписаны подряд все натуральные четные числа.
32. Составить программу, которая определяет, является ли заданное натуральное число палиндромом.

## Задания на лабораторную работу

Табл. 3.1

№	Действия (П.3)	Вывести (П.4)	Задача 1я (П.5)	Задача 2я (П.5)
1.	1	2	31	1
2.	3	8	2	5
3.	3	9	4	19
4.	2	8	8	2
5.	1	10	31	26
6.	2	4	27	8
7.	3	3	20	14
8.	3	5	10	2
9.	2	4	19	32
10.	3	4	27	31
11.	1	5	21	13
12.	3	1	23	5
13.	2	4	24	9
14.	3	7	7	1
15.	2	4	15	28
16.	2	6	4	3
17.	2	6	26	32
18.	3	4	25	1
19.	2	2	18	31
20.	2	2	32	1
21.	1	1	13	25
22.	1	6	24	17
23.	2	9	20	17
24.	2	3	8	10
25.	1	7	13	8
26.	1	2	18	19
27.	3	4	32	3
28.	1	2	29	13
29.	2	8	8	11
30.	2	5	28	20
31.	1	8	13	32
32.	1	8	6	19
33.	3	7	21	18
34.	3	6	23	17
35.	3	4	11	13
36.	3	4	5	31
37.	3	6	21	11
38.	3	9	18	1
39.	2	10	18	17
40.	1	6	7	15
41.	1	6	23	5
42.	1	4	8	22
43.	3	4	13	7
44.	1	9	22	1
45.	3	2	21	22
46.	3	8	11	30
47.	1	3	13	22
48.	2	4	7	13
49.	2	10	14	26
50.	1	6	1	18

51.	1	4	31	28
52.	3	6	28	12
53.	2	10	17	17
54.	1	6	26	19
55.	2	2	30	29
56.	2	2	27	9
57.	1	9	10	30
58.	2	10	18	13
59.	1	6	18	21
60.	1	5	14	32
61.	2	7	29	9
62.	3	8	26	21
63.	1	10	3	19
64.	2	8	6	25
65.	2	7	12	29
66.	2	3	2	20
67.	2	6	11	15
68.	2	3	1	3
69.	3	3	19	15
70.	1	10	26	8
71.	2	5	4	14
72.	2	8	19	6
73.	3	6	23	31
74.	1	8	25	32
75.	2	3	10	27
76.	3	7	30	28
77.	3	6	32	15
78.	2	3	23	25
79.	2	5	21	32
80.	1	2	18	11
81.	2	3	17	1
82.	2	1	23	11
83.	3	6	23	27
84.	2	4	1	22
85.	2	8	28	7
86.	2	8	4	18
87.	2	8	31	4
88.	3	7	32	25
89.	2	2	4	15
90.	2	1	29	15
91.	2	4	5	32
92.	3	3	25	11
93.	1	3	11	5
94.	3	5	24	28
95.	1	7	2	3
96.	2	10	22	14
97.	2	4	14	28
98.	3	4	28	21
99.	2	1	25	24
100.	2	1	4	22