Минимальные требования к практиканту. Тестирование.

Задания повышенной сложности, отмеченные звездочкой «*», необязательны.

Общие:

1. Дайте определение типа данных. Что отличает один тип данных от другого? Перечислите эти отличия с примерами.

2. Объясните, как вы понимаете операции передачи параметров в некоторую функцию «по ссылке» и «по значению». Приведите примеры и побочные эффекты каждого из вариантов.

3. *Назовите отличия интерпретируемых языков от компилируемых, перечислите достоинства и недостатки для каждого такого разделения языков.

Специализированные:

1. Назовите спецификацию, которую реализует язык JavaScript. Перечислите требования соответствия некоторого языка программирования (среди которых JavaScript и Node.js) данной спецификации (требования взять из самой спецификации). Приведите ссылку или прикрепите документ на данную спецификацию.

2. Перечислите типы данных языка JS и приведите для каждого возможные значения (для бесконечных наборов приведите по 2 примера).

3. Что вы знаете об объектах в JS? Перечислите некоторые их свойства и отличия от других типов данных в JS. Что вы знаете о прототипном наследовании?

4. * Какова максимальная длина строки в JS? Объясните подробно почему именно так.

Алгоритмы:

Для всех заданий в этой секции требуется предоставить листинг функций, написанных на JS (либо скриншот, либо читабельная вставка в виде текста-кода).

1. Напишите функцию $getMergedList(list)$, которая принимает массив $list$ произвольных числовых значений и возвращает массив, в котором все соседние значения входного массив слиты в одно.

   Примечание: значения в массиве никак не отсортированы.
   Пример:
   $[1, 2, 3, 1, 4, 5] = getMergedList([1, 1, 2, 2, 2, 3, 1, 1, 4, 5, 5])$.

2. Напишите функцию $mergeSort(list)$ или $quickSort(list)$ на выбор, сортирующую список $list$ по убыванию значений по алгоритму сортировки слиянием или быстрой сортировки соответственно.

   Примечание:
   не использовать $Array.prototype.sort(comparator)$*; аргумент функции* $list$ *есть массив чисел.*

3. Что такое сложность алгоритма? Напишите пошагово вычисление сложности для функций $getMergedList$ из задания №1 и $mergeSort(list)$ или $quickSort(list)$ (в зависимости от выбора) из задания №2.

4. Напишите функцию $computeBoolExpr$, которая принимает массив булевых значений, разделенных строками ‘and’ или ‘and’ (означают соответствующие булевы операции), и возвращает вычисленное значение на основе этого массива.

   Примечание: учитывайте приоритет операций and *и* or.
   Примеры:
   $true = computeBoolExpr([true, \text{‘}and\text{’}, false, \text{‘}or\text{’}, true\ ])$;
   $false = computeBoolExpr([false, \text{‘}or\text{’}, false, \text{‘}or\text{’}, true, \text{‘}and\text{’}, false])$.

5. * Придумайте, как модифицировать входной массив в задаче №4, чтобы оптимизировать сложность вычислений в функции $computeBoolExpr$.

6. * То же задание, что и №4, но с добавлением скобок ‘(’ и ‘)’ во входной массив для принудительного обозначения приоритета операций в виде строк перед и после true или false.

   Примечание: для оптимизации вычислений необходимо решить задачу №5.

CSS и HTML:

1. Изобразите DOM-дерево в виде блок-схемы для любого простого документа (придумайте содержимое самостоятельно).

2. Какими способами можно определить стили элемента (тэга) HTML-документа?

3. Какие типы CSS-селекторов вы знаете? Перечислите все, которые вы знаете.

4. * Что такое специфичность (specificity) CSS-селектора? Как она вычисляется и как влияет на приоритет стилей элемента?