1. Почему шестнадцатеричные числа получили широкое распространение в системном и встраиваемом программировании? В чем их преимущество перед использованием двоичных?

В системном и встроенном программировании зачастую приходится работать напрямую с аппаратными средствами. Обычно это происходит через регистры, которые могут хранить в себе состояния в виде отдельных битов (флагов). На сегодняшний день таких регистров может быть несколько тысяч в одном устройстве.

В современном мире массив двоичных (bin) чисел является слишком обширным и сложным для анализа. Для “упаковки” бинарных чисел сначала начали использовать восьмеричную систему счисления, каждая цифра которой “упаковывала” по 3 бита, затем, когда и их стало очень много, стали использовать hex-цифры (шестнадцатеричные), каждая из которых «упаковывает» по 4 бита.

1. Сколько бит занимает 1 шестнадцатеричная цифра (hex-цифра)?

4 бита: 0000 (0) – 1111 (F)

1. Сколько hex-цифер в ширину насчитывает 1 байт, 2 байта, 4 байта, 8 байт?

2, 4, 8, 16 цифры

1. Почему не рекомендовано использование целочисленных типов short, int и long во встраиваемых приложениях? Какие целочисленные типы нужно использовать и где объявлены их определения? Назови длину в байтах для каждого типа

Длина short, int и long является не фиксированной. Например, в 8битных микроконтроллерах int равен 1 байту, в 32битных – 4 байтам. Для написания более переносимого кода необходимо применять целочисленные u|int8|16|32|64\_t из stdint.h, которые являются фиксированными на всех архитектурах и ОС.

1. В чем разница между константной переменной (константой) и константным литералом (литералом)?

const int x = 5;

x – константная переменная, т.к. объявлена с модификатором const

5 – константный литерал, им инициализируется константа x. В Си все, что начинается с цифры, является литералом, поэтому имена переменных не могут начинаться с цифр

1. Каким типом по умолчанию принимается целочисленный литерал, вещественный литерал и символьный литерал (например, 'с' или '1')? Каким образом можно изменить этот тип литерала?

signed int (или просто int), double, char. Изменить можно путем явного преобразования:

unsigned short x = (unsigned short) 5;

float f = (float) 0.5;

1. В чем разница между 'a' и "a" в Си? В смысле как называется один литерал и другой и в чем их разница размещении в памяти?

'a' – символ величиной 1 байт, "a" – строка длиной 2 байта: символ 'a' и терминирующий строку нуль '\0'

1. Как использовать в Си логический (булевый) тип? С какого стандарта Си возможно его применение?

Начиная со стандарта C99 в Си доступен булевый тип \_Bool, который может принимать 0 или 1. Для привычного использования этого типа, как в C++, можно подключить stdbool.h использовать переопределение булевого типа bool, который может принимать значения 0 и 1, или false или true

1. Что обычно понимается под "магическим числом" (magic number)? Почему не рекомендовано их использование и чем обычно заменяют "магическое число" в коде?

Зачастую в коде можно встретить явное использование коэффициентов, значений регистров, адресов, и т.п., смысл которых может быть не всегда очевиден читающему код программисту. Такие числа на сленге называют магическими числами. Чтобы улучшить читаемость кода, эти явно определенные числа (неименовнные литералы) можно переопределить с помощью директивы препроцессора #define или константы. Важно то, что таким образам неименованным литералам даются осмысленные имена, которые упрощают понимание чужого кода

1. Как правильно выполнять сравнение вещественных переменных?
2. Продемонстрируй объявление с инициализацией символа и строки любым значением