1. Какие ты запомнил важные правила при написании макросов MACRO\_NAME(optional\_arg)? Чем грозит несоблюдение этих правил?

Ответ: 1) Необходимо защищать входные параметры скобками, чтобы при подстановке макроса выполнялась операция правильно. Если не защитить скобками входные параметры, можно получить неверный результат выполнения арифметических действий, либо неправильное выполнение функции

2) Защита макросов с несколькими операторами оператором do-while(0)

1. Какие преимущества и недостатки по сравнению с функцией имеет написание операции с использованием макроса?

Ответ:

Преимущества – макросы быстрее

Недостатки – функции выполняют проверку типов, а макросы просто делают замену переданного аргумента; макросы не подлежат отладке; макросы подставляют код на свое место, что приводит к большему потреблению памяти

1. В каких случаях ты бы предпочел использование макроса, а в каких – функцию?

Ответ: Макросы – при задании каких-либо констакт; при необходимости в рамках одного файла .c выполнять много раз одну и ту же операцию или функцию; использование макросов с параметрами (например, сравнение каких-то переменных или другие операции); быстрее простые операции делать макросами

Функции – при ограниченных ресурсах устройства

1. Запиши константную переменную, константный указатель на переменную какого-либо типа?

Ответ:

const float pi = 3.14f – константная переменная

char \* const str\_hello = “Hello”; - константный указатель на переменную типа char

1. Объясни словесно следующие объявления:

1)сonst char \*string;

2)const char \* const \*string\_array;

3)const int \*\* const ptr\_array;

Ответ: 1) Указатель string на константу типа char – нельзя изменить переменную по адресу, но можно изменить адрес

2) Константный указатель на константу типа char – нельзя измени ни адрес, ни переменную по адресу

3) константный указатель на указатель на константную переменную типа int – нельзя изменить адрес указателя, который указывает на другой указатель; адрес указываемого указателя менять можно, но значение переменной по этому адресу менять нельзя

1. Когда следует использовать ключевое слово const при передаче аргументов в функцию?

Ответ: Когда требуется гарантировать, что значение не будет изменяться после инициализации, Read only переменные;

1. Запиши массив со строками “Hello”, “World”, “from”, “Saint-P” через двумерный массив символов и через массив указателей на строки? В чем отличие?

Ответ: char mass[][8]: {“Hello”, “World”, “from”, “Saint-P”}; - запись через двумерный массив

char \*str[] = {“Hello”, “World”, “from”, “Saint-P”}; - запись через массив указателей на строки

Отличие в инициализации: в первом случае необходимо указывать размер максимальной строки (с учетом нулевого элемента), а также указывать многомерность массива (пустые квадратные скобки; можно указать там кол-во строк в массиве массивов)

1. Дана матрица

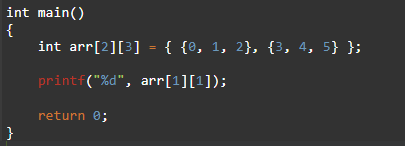
int arr[2][3] = {

{0, 1, 2},

{3, 4, 5}

};

Как обратиться к элементу с числом 4?

Ответ:

1. Что такое перечисление enum? Как объявить перечисление?

Ответ: enum имя\_перечисления {список имен} список\_переменных;

Самый первый элемент перечисления имеет порядковый номер 0, далее 1, 2 и т.д. Но можно переназначить номер имени внутри фигурных скобок, просто присвоив имени порядковый номер.

enum Gachi {

MASTER,

SLAVE,

300\_BUCKS = 300,

SO\_DEEP

};

Здесь будет такой порядок: 0, 1, 300, 301.

1. В чем разница между использованием перечислений и констант #define

Ответ: перечисления имеют свою нумерацию – это может помочь, если есть, к примеру, 20 перечислений каких-то состояний и нужно в середину вставить еще одно. В таком случае enum автоматически все пронумерует, а в случае с #define необходимо будет вручную переделывать нумерацию