1. Что такое компилятор и чем он отличается от транслятора?

Транслятор - программа, переводящая исходный код (программу, написанную на одном из высокоуровневых языков программирования) в объектный код, используемый процессором компьютера, или в промежуточный код для последующей интерпретации. Помимо осуществления перевода, трансляторы могут выявлять в исходном коде ошибки, оптимизировать исходный код, добавлять в исходный код отладочные процедуры, формировать словари идентификаторов и другое.

Компилятор – это вариант реализации транслятора, который создаётся для перевода программы, написанной на высокоуровневом языке программирования в машинный код, который в последствие будет исполняться процессором

Компилятор является видом транслятора и предназначен только для перевода кода. Осуществлять выявление ошибок в коде, оптимизацию, добавлять в исходный код отладочные процедуры и др он не способен.

1. Что такое компоновщик и какие функции он выполняет?

Компоновщик предназначен для связывания между собой объектных файлов, порождаемых компилятором, а также файлов библиотек, входящих в состав системы программирования.

Объектный файл (или набор объектных файлов) не может быть исполнен до тех пор, пока все модули и секции не будут в нем увязаны между собой. Это и делает редактор связей (компоновщик).

Таким образом компоновщик выполняет две функции:

- комбинирует (компонует, редактирует) различные объектные файлы;

- устанавливает связи между объявлением функции или переменной с ее определением.

4. Почему важен процесс проектирования ПО какие задачи входят в этот процесс?

Перед тем, как писать код, необходимо четко понимать, с какой целью он пишется. Обозначить структуру , задачи и определить пути их решения. Без подготовки процесс разработки теряет в эффективности.

Проектирование включает в себя следующие задачи:

выбор метода и стратегии решения;

выбор представления внутренних данных;

разработка основного алгоритма;

документирование ПО;

тестирование и подбор тестов;

выбор представления входных данных .

7. Какие еще важные особенности IAR workbench можно добавить в таблицу?

Возможность запуска проекта в режиме симуляции выбранного микроконтроллера.

Удобно использовать несколько конфигураций. В конфигурации Debug можно выбрать симулятор программатора, а в Release – конкретный программатор (к примеру, J-Link). Что позволит сначала отлаживать в симуляторе, не теряя время на прошивку микроконтроллера, а потом уже прошивать под конфигурацией Release.

Совместимость с большим количеством микроконтроллеров.

Мониторинг состояние процессора.

Удобная отладка с возможностью просматривать состояние переменных.

Большая база пользователей и документации.