**Q. 버퍼란 무엇이며 사용하는 이유는 무엇입니까?**

**A. 버퍼란 임시저장공간으로 입력된 값을 엔터키(\n)를 입력 받을 때 까지 저장하는 공간으로 하나하나씩 값을 전송하는 것 보다 한번에 값을 전송하는게 더 효율적이기 때문입니다.**

**Q. 변수 선언 시 주의점이 무엇입니까?**

**A.**

**-첫글자는 알파벳 또는 밑줄이 올 수 있고 그 뒤에는 추가로 숫자가 올 수 있다. (첫글자로 숫자를 쓰지 않는 이유는 C에서 8진수와 16진수 구분을 할 때 처음 시작이 숫자로 시작을 하기 때문에 오해하지 않도록 하기 위함이다.)**

**-하나의 식별자는 보통 32문자까지 짓는걸로 한다.**

**-알파벳 대소문자를 구분한다**

**-특수문자와 공백문자는 허용되지 않는다.**

**Q. 아스키코드는 왜 만들어졌습니까?**

**A. 미국에서 표준화한 것으로 정보교환을 목적으로 만들어졌다.**

**Q. 헤더파일을 선언하지 않아도 함수가 사용되는 경우는 어떤 의미입니까?**

**A. 헤더파일이 없는 경우 헤데파일 폴더를 모두 검사하여 알아서 찾아주긴 하지만 하나의 함수가 여러 헤더파일에 정의되어 있는 경우 컴파일 에러가 일어나는 경우도 있습니다.**

**Q. 함수를 선언하지 않고 main함수에서 바로 불러오고 싶다면 어떻게 해야 합니까?**

**A. 함수를 사용자 지정 헤더파일로 만들어 #include “헤더파일명”을 쓰시면 됩니다.**

**Q. 재귀함수가 무한반복이 되면 어떻게 됩니까?**

**A. 메모리중 스택부분의 호출스택이 계속 쌓이게 되면서 스택 오버플로우가 일어나게 됩니다.**

**Q. 함수 사용시 프로그램 작성의 난이도가 낮아지는 이유는 무엇입니까?**

**A. 함수를 사용할 때 분업을 하면 효율적으로 작업을 할 수가 있으며 또한 각자 맡은 함수를 조합하여 작성을 하면 프로그램의 길이가 혼자 작업할 때 보다 짧아지기 때문입니다.**

**Q. #define 지시자는 어떤 자료형을 가지나요?**

**A. 매크로함수는 자료형에 독립적이라 따로 자료형을 지정하지 않아도 알아서 자료형의 정의해서 수행합니다.**

**Q. 운영체제의 저장위치는 어디입니까?**

**A. 메인 HDD에서 구동이 되며 하드디스크 부팅섹터라는 부분에 정보가 저장됩니다.**

**Q. 포인터 연산에서 곱셈과 나눗셈이 가능합니까?**

**A. 안됩니다. 왜냐하면 메모리 주소에 대한 연산은 덧셈과 뺄셈만 허용되며 메모리 주소 연산에 곱셈과 나눗셈은 필요가 없기 때문입니다.**

**Q. 컴파일 되는 과정을 설명해 주십시오.**

**A. 선행처리기가 실행이 되고 원시프로그램이 생성이 된 후 컴파일러가 기계어로 번역하여 목적 프로그램을 생성한 다음 링커를 통해 라이브러리 파일과 목적 프로그램을 합친 다음 실행파일을 만들게 됩니다.**