**[문제1]** 키워드 const의 의미 (const는 그 대상을 변경하지 않는 "상수"를 의미.)

키워드 const는 어떠한 의미를 갖는가? 다음 문장들을 대상으로 이를 설명해보자.

const int num = 10; → 변수 num을 상수화!

const int \* ptr1 = &val1; → 포인터 ptr1을 이용해서 val1의 값을 변경할 수 없음.

int \* const ptr2 = &val2; → 포인터 ptr2 상수화!

const int \* const ptr3 = &val3; → 포인터 ptr3 상수화 && 포인터 pt3이용해서 val3의 값 변경 못함.

**[문제2]** 실행중인 프로그램의 메모리 공간

실행중인 프로그램은 운영체제로부터 메모리 공간을 할당 받는데, 이는 크게 데이터, 스택, 힙 영역으로 나뉜다. 각각의 영역에는 어떠한 형태의 변수가 할당되는지 설명해보자. 특히 C언어의 malloc과 free함수와 관련해서도 설명해보자.

**데이터(Data)영역(전역변수, static변수)**

→ 전역 변수와 static 변수가 할당되는 영역

→ 프로그램의 시작과 동시에 할당되고, 프로그램이 종료되어야 메모리에서 소멸됨  
 **스택(Stack) 영역(지역변수, 매개변수, 컴파일 타임에 크기 결정)**

→ 함수 호출 시 생성되는 지역 변수와 매개 변수가 저장되는 영역

→ 함수 호출이 완료되면 사라짐  
  
**힙(Heap) 영역(프로그래머 할당, 런 타임에 크기 결정)**

→ 필요에 의해 동적으로 메모리를 할당할 때 사용

→ 할당해야 할 메모리의 크기를 프로그램이 실행되는 동안 결정해야 하는 경우(런 타임 때) 유용하게 사용되는 공간

**malloc & free**

malloc 함수호출에 의해 할당된 메모리 공간은 free 함수호출을 통해서 소멸하지 않으면 해제되지 않는다.

**[문제3]** Call-by-value vs. Call-by-reference

함수의 호출형태는 크게 ‘값에 의한 호출(Call-by-value)’과 ‘참조에 의한 호출(Call-by-reference)’로 나뉜다. 이 둘을 나누는 기준이 무엇인지, 두 int형 변수의 값을 교환하는 Swap 함수를 예로 들어가면서 설명해보자.

→

Void SwapByValue(int num1, num2){

int temp = num1;

num1 = num2;

num2 = temp;

} // Call-by-value

Void SwapByRef(int \* ptr1, int \* ptr2){

Int temp = \*ptr1;

\*ptr1 = \*ptr2;

\*ptr2 = temp;

} // Call-by-reference