**Malloc과 Free**

**malloc 함수**

동적 메모리 할당은 프로그램 실행 중에 동적으로 메모리를 할당하는 것을 의미한다. 동적할당을 사용하는 이유는 메모리 낭비를 줄이기 위함이다. 여기서 동적이란 가변적을 의미한다. 그리고 중요한 것은 동적으로 메모리 할당을 진행할 때, 힙 영역을 사용한다.

(힙(Heap)영역은 메모리 구조 중 하나로 사용자가 직접 관리할 수 있는 메모리 영역이다. 힙영역은 메모리의 낮은 주소에서 높은 주소의 방향으로 할당된다.)

출처 : <http://tcpschool.com/c/c_memory_structure>

동적 메모리 할당을 해주는 함수는 malloc이다. Malloc은 **memory allocation**의 약자이다. 이 함수는 <stdlib.h>에 속해있기에 전처리기에 #include<stdlib.h>로 헤더파일을 열고 사용해야 한다.

arr = (int \*)malloc(sizeof(int) \* SizeOfArray);

위 코드는 int 형인 배열을 가변적인 크기 SizeOfArray로 int 형 배열인 arr의 메모리 공간을 할당하는 코드이다.

여기서 이 함수가 하는 역할은 인자로 전달된 크기의 바이트 수만큼 메모리 공간을 만든다. 이 함수의 반환값이 할당한 메모리의 시작주소를 의미한다. 이때 이 시작주소 값은 void \* 형태이기 때문에 int \*로 형변환을 하여 arr에 값을 넣는 것이다.

**free 함수**

마지막에 free함수는 메모리 공간을 할당받고 사용한 후 메모리 영역을 다시 컴퓨터에게 돌려주는 역할을 한다. 이를 “해제(free)한다” 라고 칭한다. 이 free 함수로 메모리 공간을 반환하지 않는다면 딱히 사용하지도 않는 메모리가 쓸데없이 자리만 차지한다. 그렇게 된다면 메모리의 효율적인 사용을 위한 malloc을 사용하는 이유가 반감되므로 무조건 free함수를 사용해야한다. 이러한 메모리 낭비현상을 메모리 누수(memory leak) 이라고 한다.

출처 : <https://modoocode.com/98>

**Node의 구현에서 사용한** node \*newNode = (node \*)malloc(sizeof(node)); **의 의미**

(node \*)malloc(sizeof(node));는 node구조 체만큼 메모리를 할당하라는 뜻이다. 그리고 구조체에서 사용하면 sizeof(구조체)는 구조체 속의 자료형의 합이 된다.

typedef struct \_node {

char value; //char형의 자료크기는 1 byte (32비트 기준)

struct \_node \*next; //포인터 형 자료의 크기는 4 byte (32비트 기준)

} node;

위 구조체에서 sizeof(node)는 구조체 속의 자료형의 합인 5byte가 된다.