

Stack 의 push pop 이 이루어지는 루틴 정리.

1. 구조체를 만들어 준다. 안에는 malloc 으로 heap 공간을 만들시 필요한 구성품들을 넣어준다.
  - 1) data : stack 에 쌓여 이동하는 것을 보아야하기 때문에 data 가 들어간다.
  - 2) struct \_\_stack \*link : 구조체형 포인터 변수인 링크를 만든다.
2. data 간의 링크 노드를 만들어 그 주소를 반환해 주는 함수를 만든다. (get\_node)
  - 1) Stack 형 포인터 함수를 만든다.
  - 2) Stack 형 포인터 변수를 만들어 malloc 함수를 이용해 heap 공간을 생성한다.
  - 3) 생성되는 link 의 0 값을 넣는다.
  - 4) tmp 가 리턴 된다.
3. Push 라는 함수를 만든다.
  - 1) Stack 형 포인터 함수 만든다.
  - 2) push 할 데이터와 포인터변수 주소를 받는다.
  - 3) 받은 주소에 get\_node 를 이용해서 heap 공간을 생성한다.(구조체 블록을 불러옴)
  - 4) 새로 생성한 구조체 블록에 data 값을 저장한다.
  - 5) 새로 생성한 구조체 블록의 link 값에 현재 주소값을 저장한다.

(Stack \*tmp 를 하는 이유는 임시적으로 현재 heap 공간을 가지고 있기 위해서 선언을 해주는 것이다.)  
tmp 의 용도는 임시 저장소 같은것이고 우리가 링크시 변화시켜주는 놈은 \*top, \*head, \*root 인 것이다.

->한마디로 tmp 는 1)공간 할당 후 고정 2) 공간해제를 위해서 쓰는 것이다.