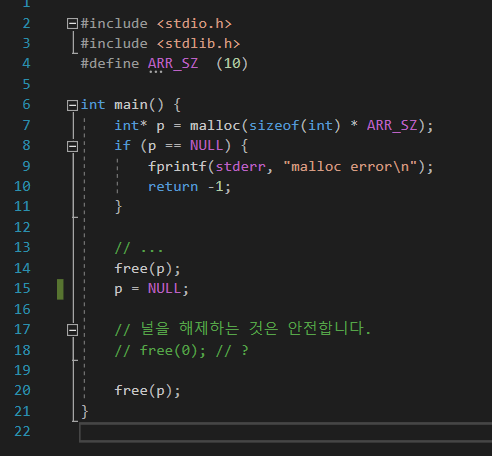


같은 레이어에서 해제 해줘야하지 위 코드는 한단계 더 들어간 레이어에서도 해제, 즉 2번의 해제를 했기 때문에 미정의 동작이다.

->동일한 추상화 레벨의 같은 모듈에서 메모리를 할당하고 해제하는 것이 좋다.

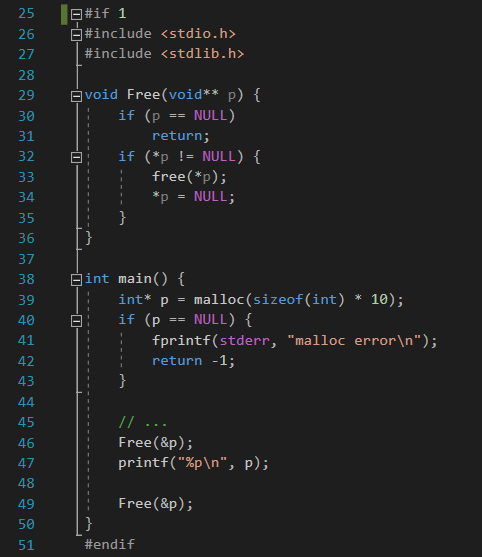


free(p); 를 두번 해주게 되면 오류가 뜬다. 그것을 해결하기 위해

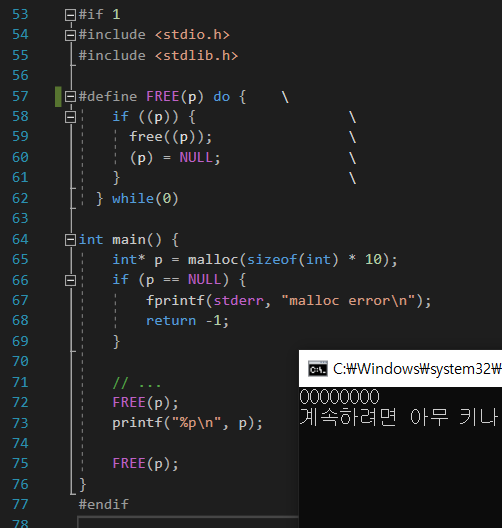
포인터를 다 사용했으면 p = null;을 해준후, free(0);을 해주자. 그러면 free(p); -> p = null; -> free(p);가 가능해진다.

그러나 일일이 사용자가 p를 null로 해주는 것은 굉장히 번거롭다.

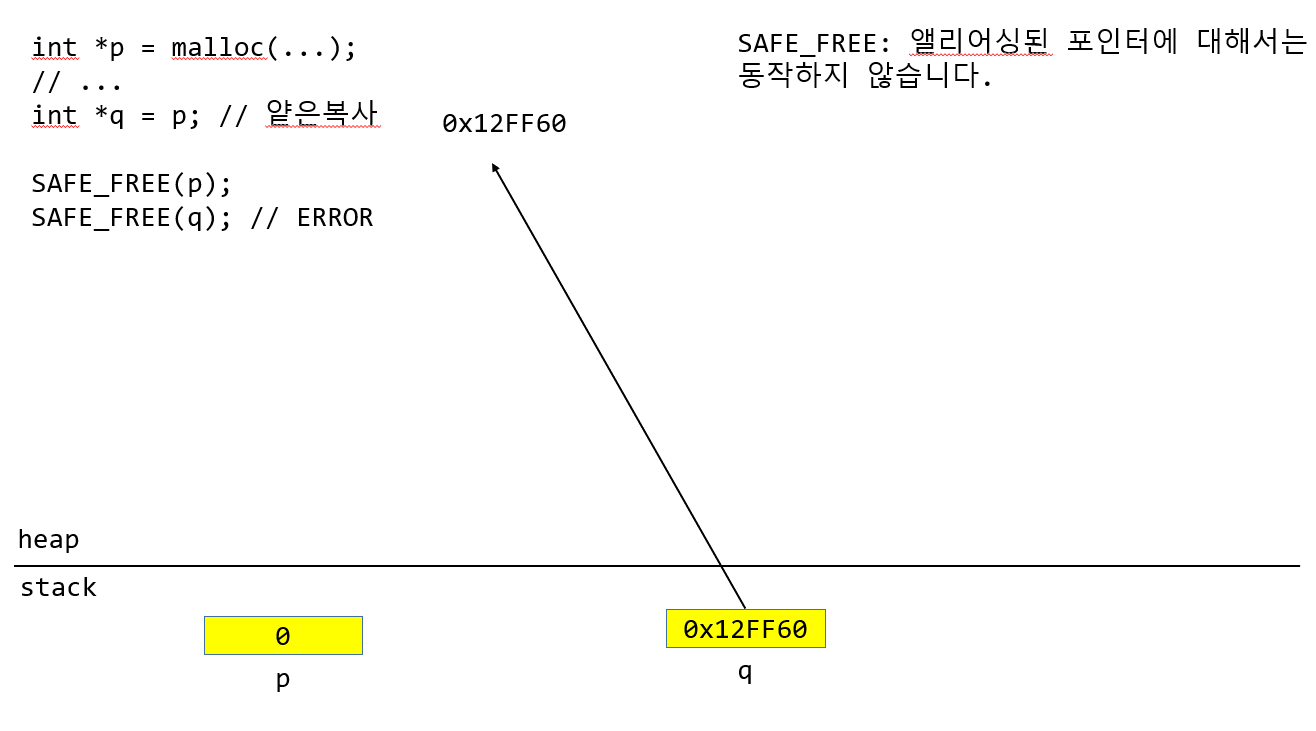
그래서 해제함과 동시에 p를 null로 바꿔주는 함수를 만들어주면 편할것이다.



void \*(모든 타입받는것의) \*(주소)p

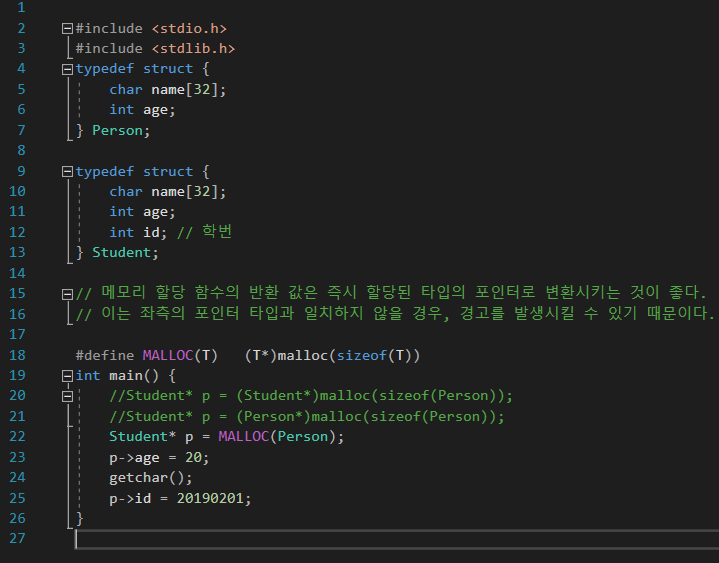


이전 코드에서 만든 함수를 메크로로!

위 코드에서의 한가지 제약조건(단점)

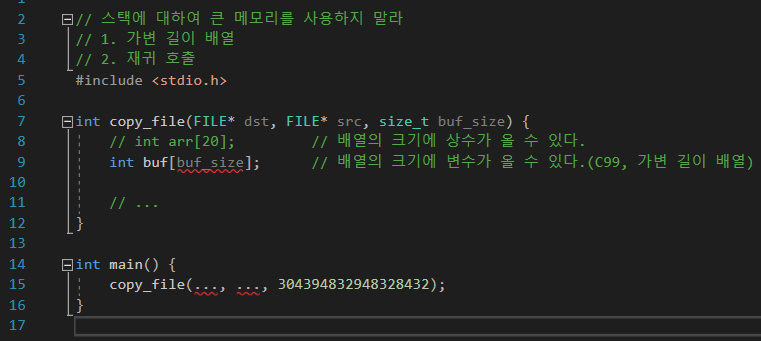
얕은복사를 하게되면 주소의 정보만 복사되므로 p가 가르키는 것을 지우고 q가 가리키는 것을 지우려하면 주소의 정보를 통해 if문은 통과하지만 이미 지워진 것을 또 free하려고 하므로 오류가 뜬다.

(cf. 같은 포인터가 같은곳을 가르키는 것을 ‘엘리어싱된 포인터’라고한다.)

malloc하게 되면 void\*형이다. 이를 Person \* 로 캐스팅해라! (인자로 전달된 타입에 대해 캐스팅!!)

그렇게 되면 왼항과 우항이 호환이 안되면 적어도 컴파일러가 경고는 내준다

->메모리 할당 함수의 반환값은 즉시 할당된 타입의 포인터로 변환시키는 것이 좋다. 이는 좌측의 포인터 타입과 일치하지 않을 경우, 경고를 발생시킬수 있기 때문이다.

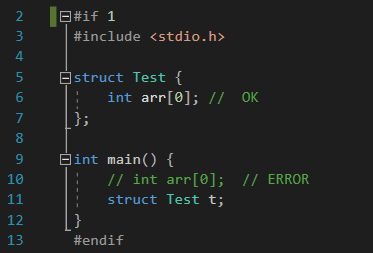


1. 가변길이 배열

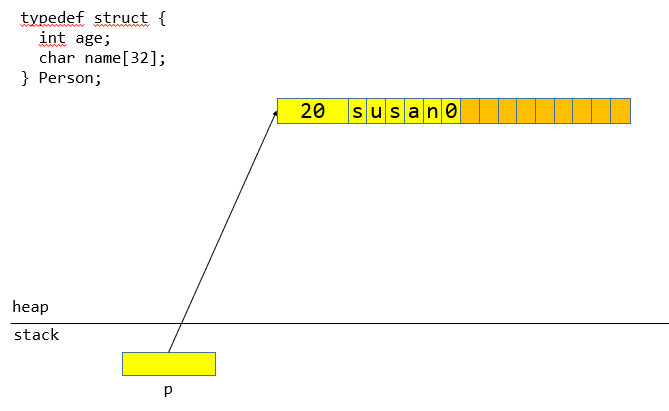
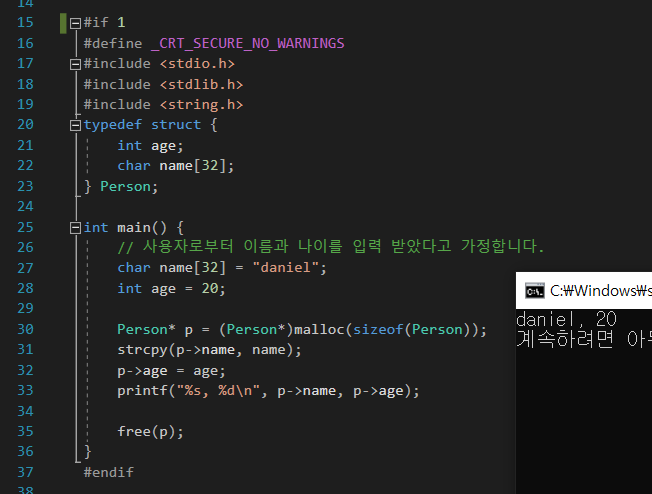
2. 재귀 호출

배열의 크기에 상수가 올수 없다.

배열의 크기에 변수가 올 수 있다.(c99, 가변길이 배열)

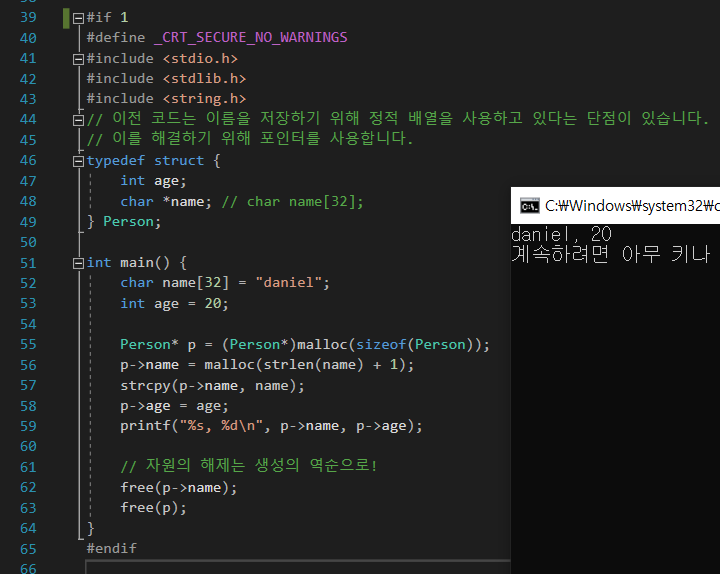


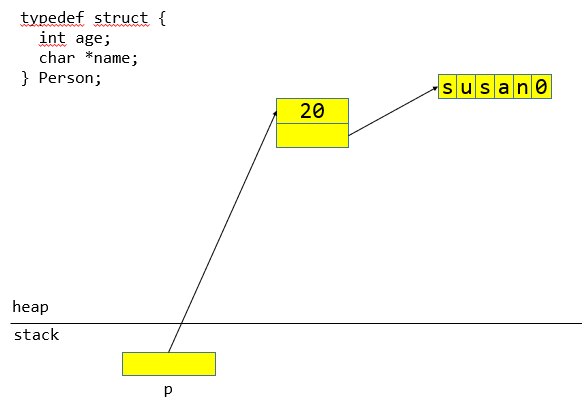
struct안에서의 int arr[0];은 되는데 왜 main에서의 int arr[0]; 은 안되나?



이 코드의 단점은? – 이름을 저장하기 위해 정적배열을 사용하고 있어 낭비되는 메모리가 생김

이를 해결하기 위해 포인터 사용(아래코드)

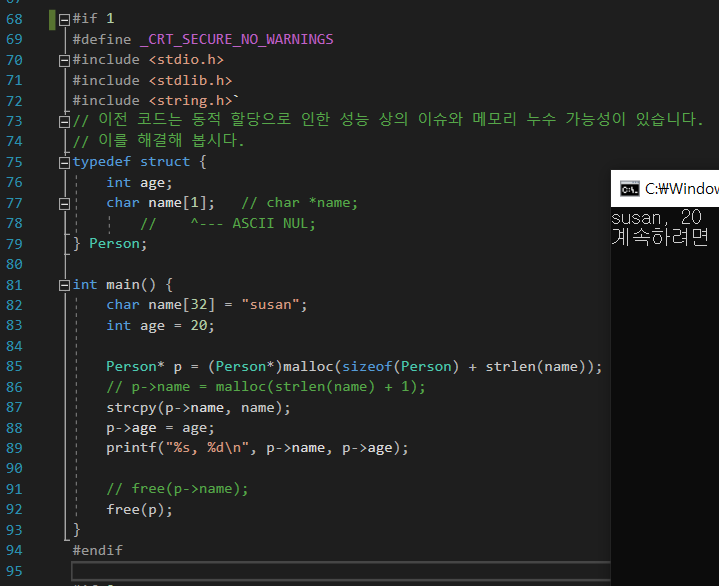


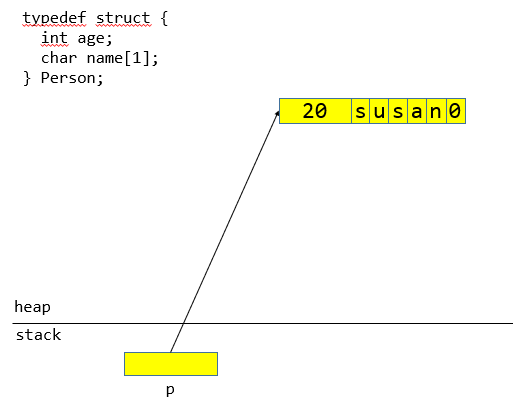


단점:

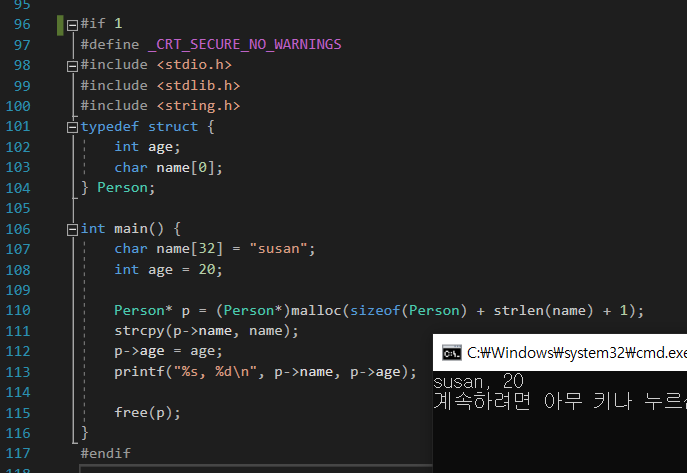
1. 동적할당은 굉장히 큰 작업인데 그것을 두번이나 하고있다.(즉, 성능상의 이슈)
2. 만약 free(p->name);을 실수로 안하게되면 힙 메모리 누수가 발생하고 나중에 시간이 계속해서 지나면 프로그램은 죽을것이다.

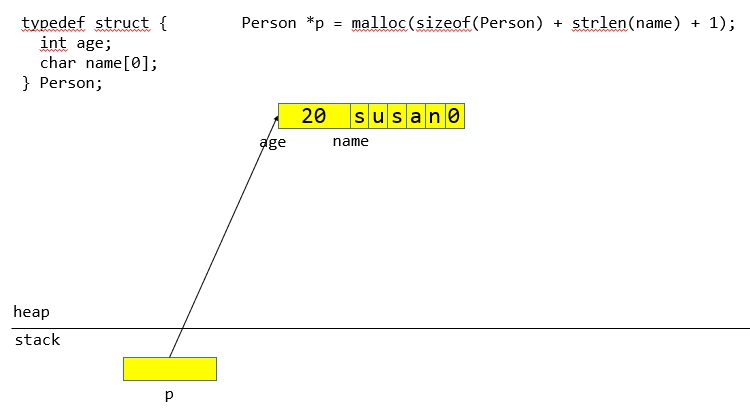
->즉 이 코드는 동적할당으로 인한 성능 상의 이슈와 메모리 누수 가능성이있다.





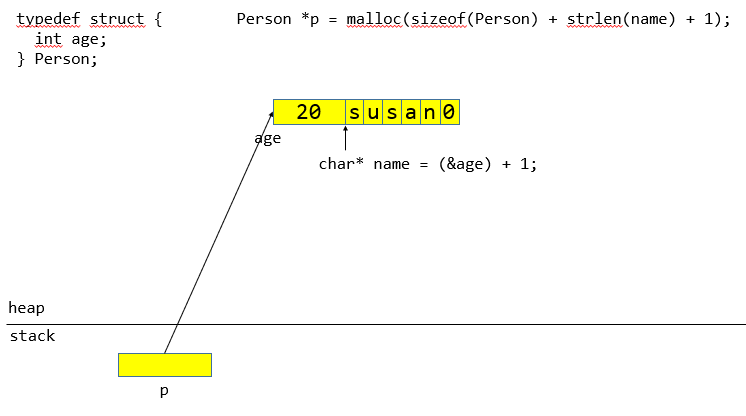
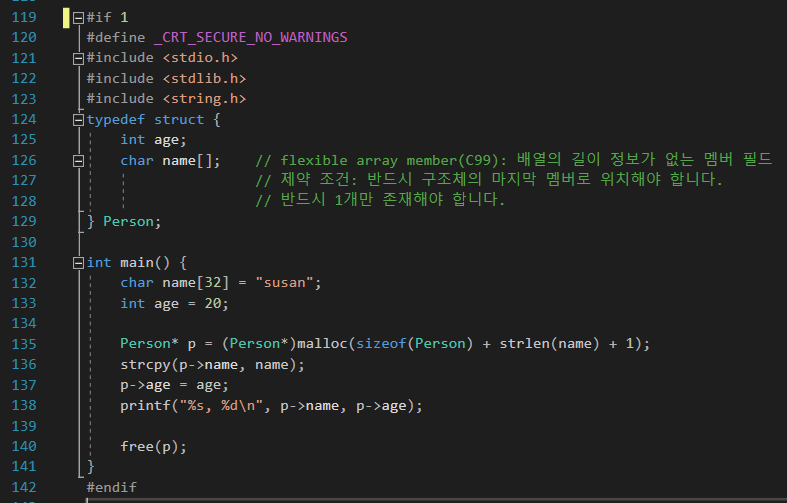
어차피 name에 대한 것은 주소만 존재하므로 그것을 복사하고 배열1칸에는 \0만 넣을수 있게한다.





아스키 널을 고려할필요 없게 하기위해서

그러나 이것은 한가지 문제가 존재한다. -> 길이가0인 배열은 c표준에 위배된다.. 표준이 아니기 때문에 동작하지 않는 컴파일러가 있다. 그래서 표준위원에서는 정식으로 채택햇는데 그 방법으로는 아래코드이다

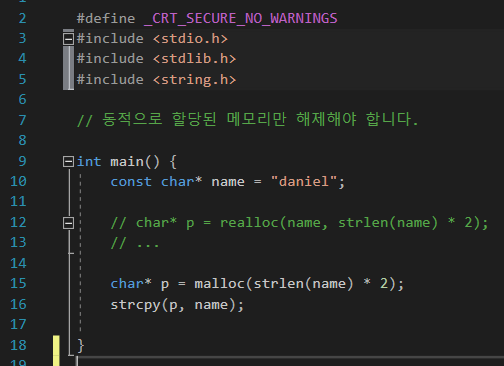


char name[]; 위처럼 0을 쓰는 것이 아니라 그냥 아무것도 안쓴다. -> 배엵의 길이 정보가 없는 멤버 필드(flexible array member(c99)) (이것의 제약조건: 반드시 구조체의 마지막 멤버로 위치해야)

만약 중간에 들어가면 sizeof(person)햇을 때 정상적인 값이 나오지 않는다.

즉, 이러한 길이정보없는 멤버필드는 반드시 1개만 존재해야한다.

* 위의 예제들보단 지금 이예제처럼 flexible array member)를 사용하자



realloc은 name을 복사하고 p에 전달한 다음 name을 해제하게 되는데 heap이 아닌

stack에 할당된 부분에 대해서 해제를 하고있다.. (오류!)

* 즉 동적으로 할당된 메모리만 해제해야한다.