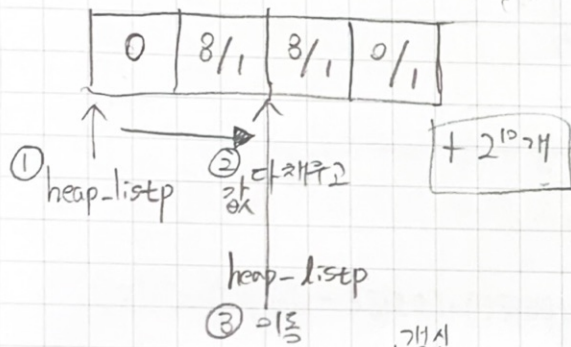
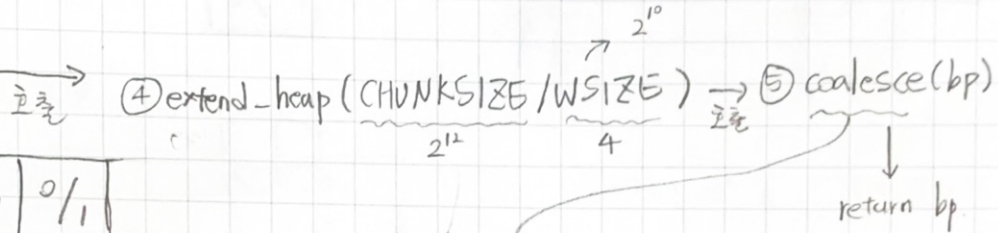


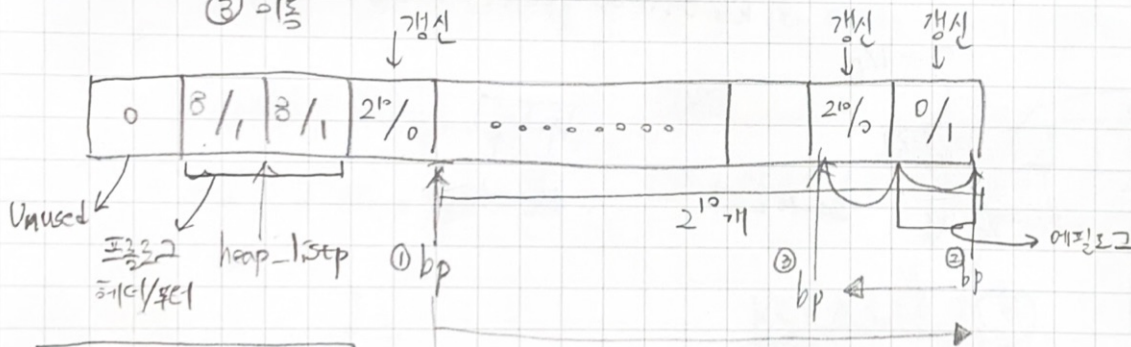
mem_init()



mm_init()



현재 bp가 가리키는 블록의 이전 블록과 다음 블록의
항상 여부를 확인하여 가용 블록이 있다면
현재 블록과 인접 가용 블록은 하나로 합친다.



mm_malloc(14)

14를 8의 배수로 만들

⇒ 16

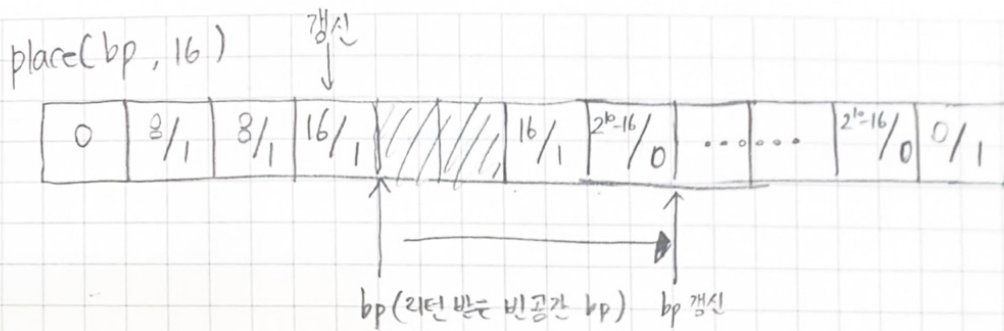
$bp = \text{find_fit}(16)$ // 16크기를 할당할 수 있는 메모리 주소를 찾는다.

heap-listp를 움직인다 그리고

bp에 heap-listp를 담는다

이 bp의 헤더의 ALLOC 값을 확인해서 free 이고 16이 헤더에있는 데이터의 크기와 비교해서 작으면
할당할 수 없으니까 return bp를 한다. 만약 조건을 만족하지 않으면 bp를 다음 블록 포인터로 이동한다.
이동해서 빈 블록을 찾는 것이다. 찾으면 그 위치의 블록포인터 (bp)를 반환한다. (아직 값 할당 안함)

↓
 $\text{place}(bp, 16)$ // 값 할당 함수



메모리에 할당받을 공간이 있다면

$$\text{extendsize} = \text{MAX}(16, \underbrace{2^{12}}_{\text{CHUNKSIZE}}) = 2^{12}$$

$$bp = \text{extend_heap}(\underbrace{2^{12}/4}_{\substack{\downarrow \text{WSIZE} \\ 2^{10}}}) \quad // \text{힙공간 자체를 늘려준다.}$$

이후 place(bp, 16) 을 호출하여 값을 할당

결과적으로 mm_malloc(14) 는 위와 포인터 bp를 리턴한다.