

int A[], *B; // 看哪个不正确.

① B = &(*A)
 ② B = A + *B
 ③ *A = A[1]
 ④ B = *(*A)

Question.

① B = &(*A)
 ↓
 是个数(int)
 ↓
 一个数的地址(int)
 ∴ B 是 int 的地址
 ∴ B = &(*A) ✓

④ B = *(*A)
 ↓
 这是地址
 ↓
 这是 A[0] 是个数(int)
 ↓
 一个数的 dereference ???
 这两个相等 ??? X

② B = A + *B
 ↓
 这个 B 上存的数(int)
 ↓
 A + *B 可以表示为
 A[*B] 的地址(int)
 ↓
 它是个 int.
 ∴ B 是 int 的地址
 ∴ B = A + *B ✓

eg = A = {0, 2, 4, 3, 4, 8, 7}

首先 A 是个 array, 所以 A 本身是个 pointer.

⇒ *A 表示 A[0] 的数值 eg = *A = A[0] = 0.
 B 的 type 是 int*, 所以 B 是个 pointer.

注: 在声明的时候可以这样读:

// 声明
 int *B;
 ↓
 数据类型是 int type 的指针!

放在之后的 B 就是个地址.

PS: 一开始可以理解为
 int *B
 是个 int 也就是整数,
 ∴ * 表示 dereference
 ∴ 所以 B 是个位置, 可以找回这个位置存放的数据

③ *A = A[1]
 *A 表示 array A 的第一个数的值也就是 A[0].
 ∴ *A 是个数, 且 A[1] 是个数
 ∴ *A = A[1] ✓

注: 这句话相当于令 A[0] = A[1] eg = A = {2, 2, 4, 3, 4, 8, 7}