	주소	데이터
	132	
int a;	131	
	130	?
int b;	129	4 bytes int
int c;	128	
The c,		
	107	
	106	?
	105	4 bytes int
	104	
	103	
	102	?
	101	4 bytes int
*	100	
	099	

- 위의 메모리 주소들은 이해를 돕기 위한 쉬운 예시이며 실제와는 차이가 있습니다.
- 디버거를 이용해서 메모리를 직접 들여다보는 방법은 이후 강의에서 자세히 다릅니다.

	주소	데이터
	132	
<pre>int a;</pre>	131	
total by	130	7
<pre>int b;</pre>	129	4 bytes int
<pre>int c;</pre>	128	
THE C,		
- 7.0	107	
a = 7; •	106	8
b = 8;	105	4 bytes int
	104	
c = a + b;	103	
	102	15
	101	4 bytes int
	100	
	099	
. 이이 메ㅁ리 조스	드이 이체로 도기 이치	신으 에시이머 시계(

네이터	수소
	132
	131
7	130
4 bytes int	129
	128
	107
8	106
4 bytes int	105
	104
	103
15	102
4 bytes int	101
1	100
	099

- 위의 메모리 주소들은 이해를 돕기 위한 쉬운 예시이며 실제와는 차이가 있습니다.
- 디버거를 이용해서 메모리를 직접 들여다보는 방법은 이후 강의에서 자세히 다릅니다.



int a = 7;

int형 변수의 <mark>주소</mark>를 저장하는 변수 주소 연산자 Address-of operator

int \*a\_ptr = &a;

\*: asterisk

주소	데이터
132	
131	
130	7
129	4 bytes int
128	
107	
106	128
105	4 bytes int*
104	

x86에서는 4바이트 x64에서는 8바이트

포인터 변수에 저장되어 있는 값은 다른 변수의 주소이다.

따라서 직접적으로 데이터를 저장하기 보다는

다른 메모리 공간이나 그 공간에 저장되어 있는 데이터(또는 객체Object)를 (간접적으로) 가리키는 역할을 한다.



- 위의 메모리 주소들은 이해를 돕기 위한 쉬운 예시이며 실제와는 차이가 있습니다.
- 디버거를 이용해서 메모리를 직접 들여다보는 방법은 이후 강의에서 자세히 다룹니다.

int 
$$a = 7$$
;

\*a\_ptr가 int 자료형이라는 의미

간접 접근Indirection 역참조dereferencing 방향 재지정redirection

주소	데이터
132	
131	
130	<sub>-</sub> 8
129	4 bytes int
128 🕟	
107	
106	→ 128
105	4 bytes int*
104	

x86에서는 4바이트 x64에서는 8바이트

- 위의 메모리 주소들은 이해를 돕기 위한 쉬운 예시이며 실제와는 차이가 있습니다.
- 디버거를 이용해서 메모리를 직접 들여다보는 방법은 이후 강의에서 자세히 다룹니다.



int a = 7;
int \*a\_ptr = &a;
\*a\_ptr = 8;

int c = 9;

c += \*a\_ptr;

주소	데이터
132	
131	
130	. 8
129	4 bytes int
128 😞	
107	
106	<b>128</b>
105	4 bytes int*
104	
103	
102	17
101	4 bytes int
100	-
099	

x86에서는 4바이트 x64에서는 8바이트

- 위의 메모리 주소들은 이해를 돕기 위한 쉬운 예시이며 실제와는 차이가 있습니다.
  - 디버거를 이용해서 메모리를 직접 들여다보는 방법은 이후 강의에서 자세히 다룹니다.