Pointers to pointers to pointers to pointers

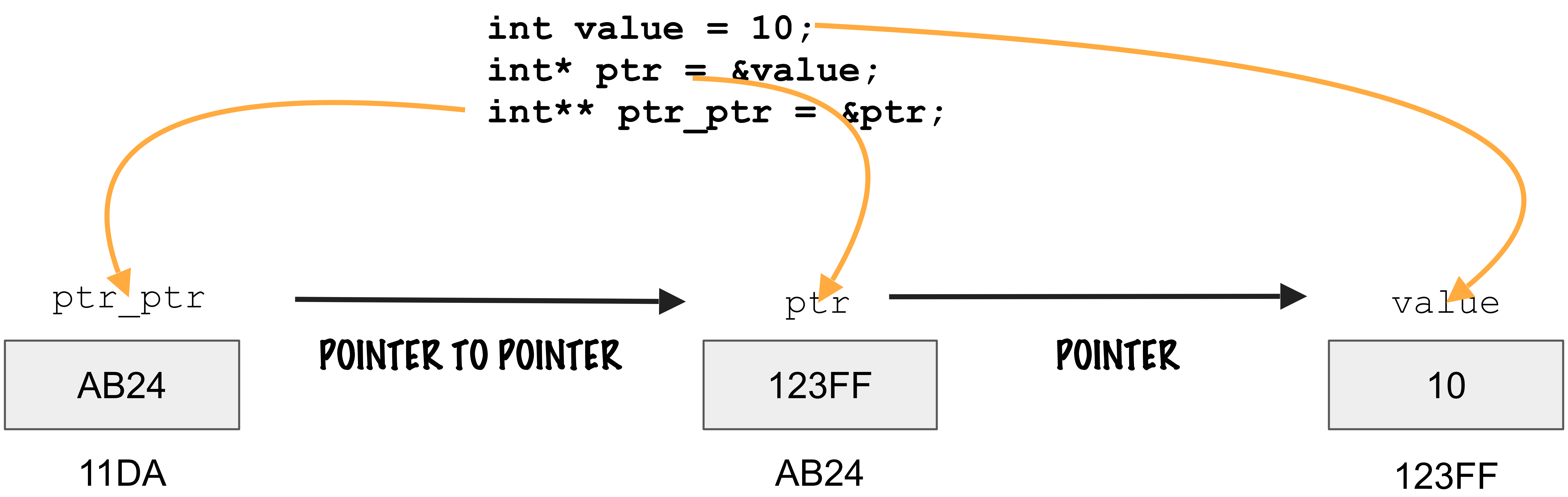
We’ve learnt that pointers are just holders of memory addresses, memory at that location holds the value.

This value can be of any type and determines the type of the pointer.

Pointers can also point to other pointers, i.e. hold the memory location of another pointer which holds the memory location of a value.

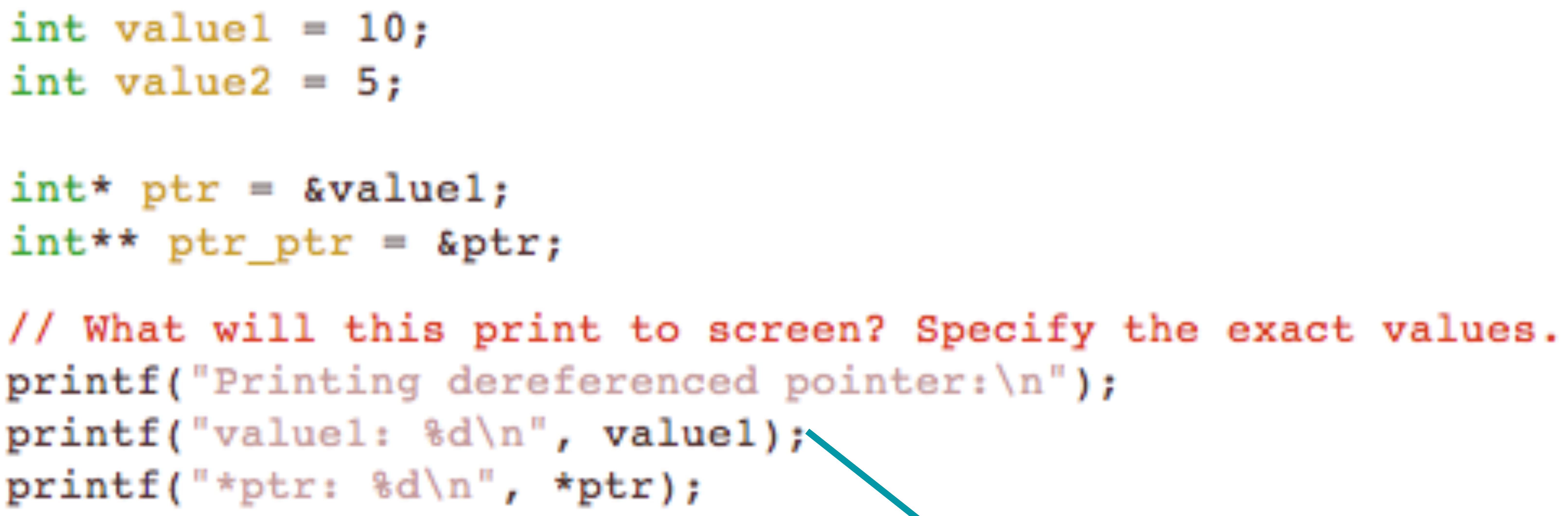
This pointing can be nested to any number of levels

POINTERS TO POINTERS



```
int value = 10;
int* ptr = &value;
int** ptr_ptr = &ptr;
```

POINTERS TO POINTERS - BEND YOUR MIND, STARTING SLOW



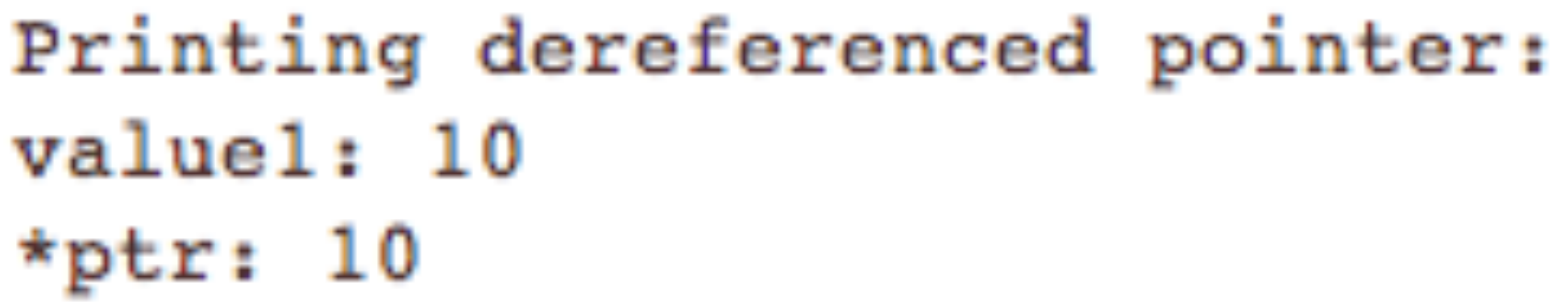
```
int value1 = 10;
int value2 = 5;

int* ptr = &value1;
int** ptr_ptr = &ptr;

// What will this print to screen? Specify the exact values.
printf("Printing dereferenced pointer:\n");
printf("value1: %d\n", value1);
printf("*ptr: %d\n", *ptr);
```

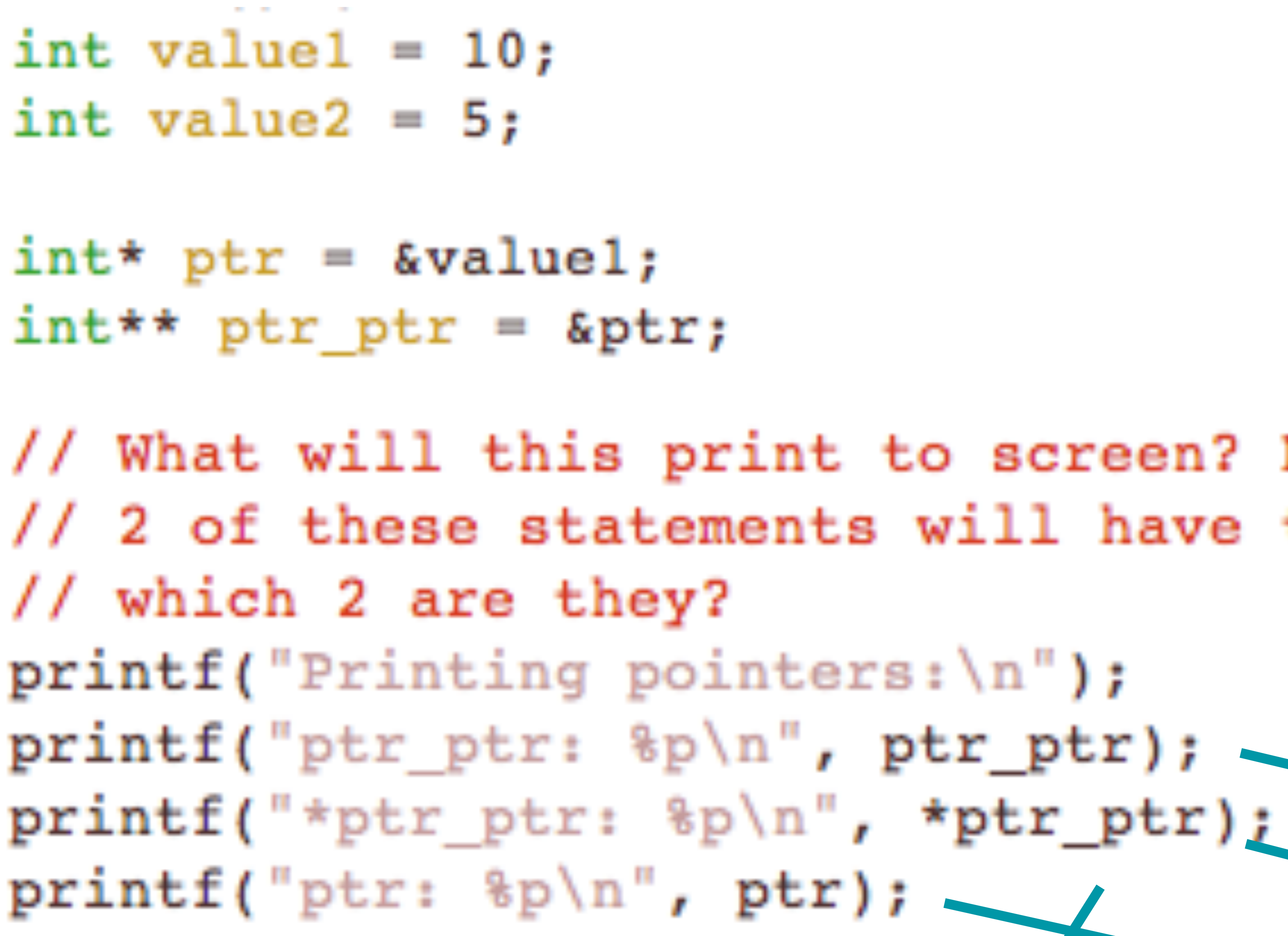
HINT: SIMPLE ONE LEVEL DEREFERENCING OF A POINTER

ANSWER



```
Printing dereferenced pointer:
value1: 10
*ptr: 10
```

POINTERS TO POINTERS - BEND YOUR MIND

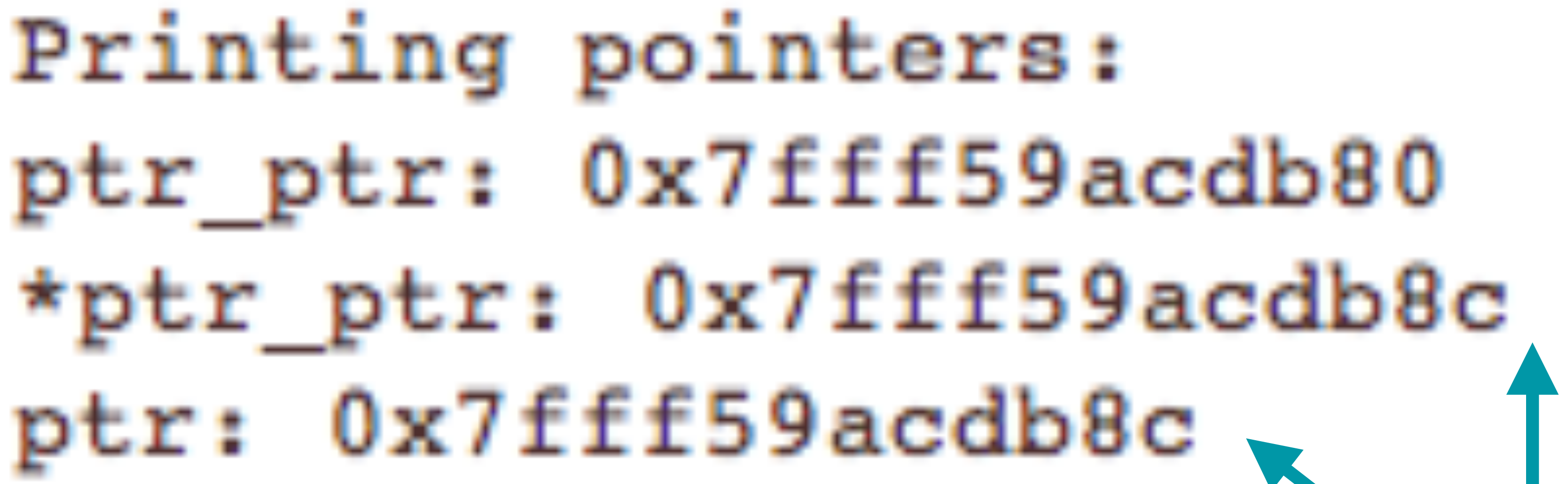


```
int value1 = 10;
int value2 = 5;

int* ptr = &value1;
int** ptr_ptr = &ptr;

// What will this print to screen? Exact values are not needed.
// 2 of these statements will have the same value printed to screen,
// which 2 are they?
printf("Printing pointers:\n");
printf("ptr_ptr: %p\n", ptr_ptr);
printf("*ptr_ptr: %p\n", *ptr_ptr);
printf("ptr: %p\n", ptr);
```

ANSWER

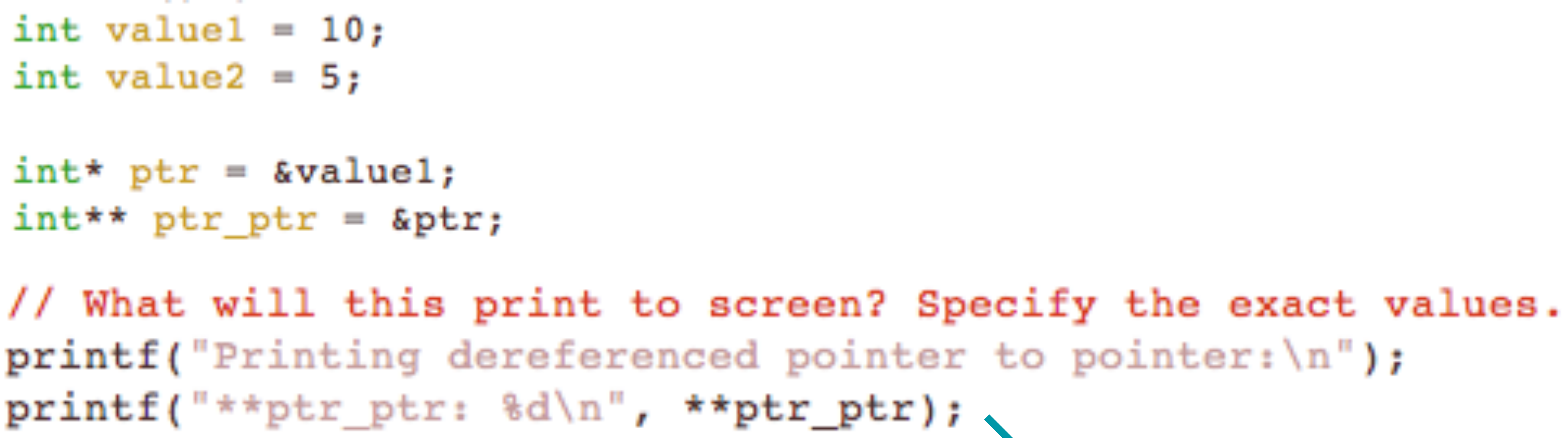


```
Printing pointers:
ptr_ptr: 0x7fff59acdb80
*ptr_ptr: 0x7fff59acdb8c
ptr: 0x7fff59acdb8c
```

HINT: THIS IS A POINTER TO A POINTER WHICH MEANS DEREFERENCING IT ONCE MEANS ITS A POINTER

THESE 2 ADDRESSES ARE THE SAME SINCE THEY REFER TO THE SAME POINTER (PTR)

POINTERS TO POINTERS - MORE MIND BENDING, SPEEDING UP

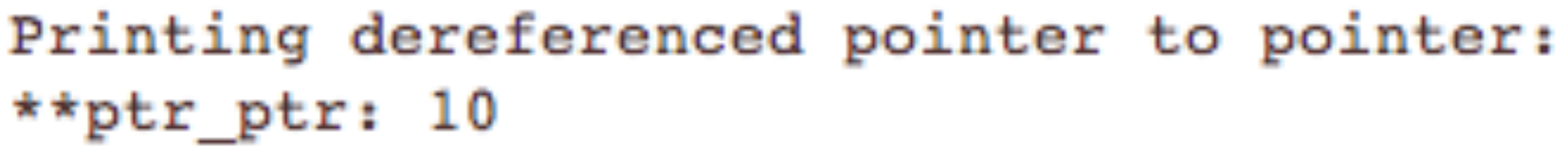


```c
int value1 = 10;
int value2 = 5;

int* ptr = &value1;
int** ptr_ptr = &ptr;

// What will this print to screen? Specify the exact values.
printf("Printing dereferenced pointer to pointer:\n");
printf("**ptr_ptr: %d\n", **ptr_ptr);
```

ANSWER



```
Printing dereferenced pointer to pointer:
**ptr_ptr: 10
```

HINT: THIS IS A POINTER TO A POINTER WHICH MEANS DEREFERENCING IT TWICE MEANS ITS AN INTEGER