Recap

Recap

Variable

Recap

Variable	`int a`

Recap

Variable	`int a`
Address of `a`	

Recap

Variable	`int a`
Address of `a`	`&a`

Recap

Variable	`int a`
Address of `a`	`&a`
Pointer	

Recap

Variable	`int a`
Address of `a`	`&a`
Pointer	`int *p`

Recap

Variable	`int a`
Address of `a`	`&a`
Pointer	`int *p`
Pointer to `a`	

Recap

Variable	`int a`
Address of `a`	`&a`
Pointer	`int *p`
Pointer to `a`	`p = &a`

Recap

Variable	`int a`
Address of `a`	`&a`
Pointer	`int *p`
Pointer to `a`	`p = &a`
Access `a`	

Recap

Variable	`int a`
Address of a	`&a`
Pointer	`int *p`
Pointer to a	`p = &a`
Access a	`a = 5`

Recap

Variable	`int a`
Address of a	`&a`
Pointer	`int *p`
Pointer to a	`p = &a`
Access a	`a = 5 or *p = 5`

int a

int a

int *pa

int a

int *pa

float b

int a

int *pa

float b

float *pb

int a

int *pa

float b

float *pb

char c

int a

int *pa

float b

float *pb

char c

char *pc

int a

int *pa

float b

float *pb

char c

char *pc

```
void swap(int *pa, int *pb)
{
  int t;
  t = *pa;
  *pa = *pb;
  *pb = t;
}
```

int a	int *pa
float b	float *pb
char c	char *pc

```
void swap(float *pa, float *pb)
{
  float t;
  t = *pa;
  *pa = *pb;
  *pb = t;
}
```

int a	int *pa
float b	float *pb
char c	char *pc

```
void area_peri(int len, int bth, int *ar, int *pr)
{
  *ar = len * bth;
  *pr = 2 * (len + bth);
}
```

int a	int *pa
float b	float *pb
char c	char *pc

```
void area_peri(int len, int bth, int *ar, int *pr)
{
  *ar = len * bth;
  *pr = 2 * (len + bth);
}
```

int a	int *pa
float b	float *pb
char c	char *pc

```
void area_peri(float len, float bth, float *ar, float
*pr)
{
  *ar = len * bth;
  *pr = 2 * (len + bth);
}
```

int

int

int *

int a

int *pa

int a

int *pa

float

int a

int *pa

float

float *

int a

int *pa

float b

float *pb

int a

int *pa

float b

float *pb

char

int a

int *pa

float b

float *pb

char

char *

int a

int *pa

float b

float *pb

char c

char *pc

```
int a;
```

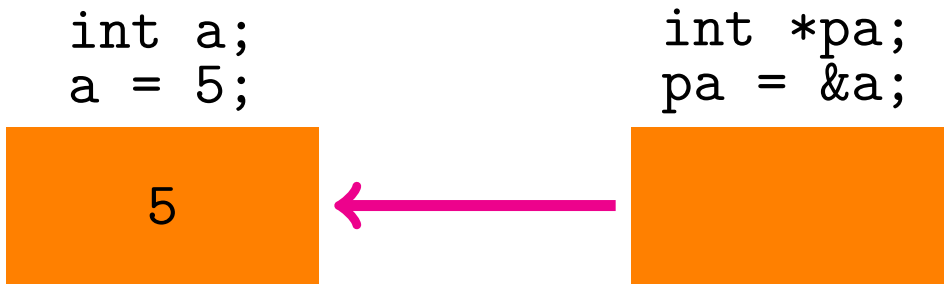
```
int a;
a = 5;
```

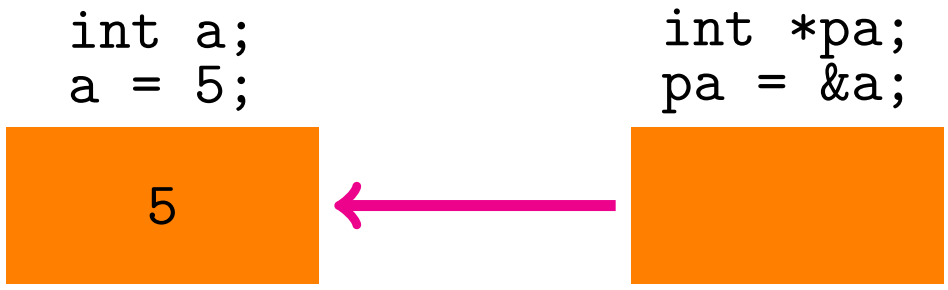
```
int a;
a = 5;
```


```
int *pa;
```

```
int a;
a = 5;
```

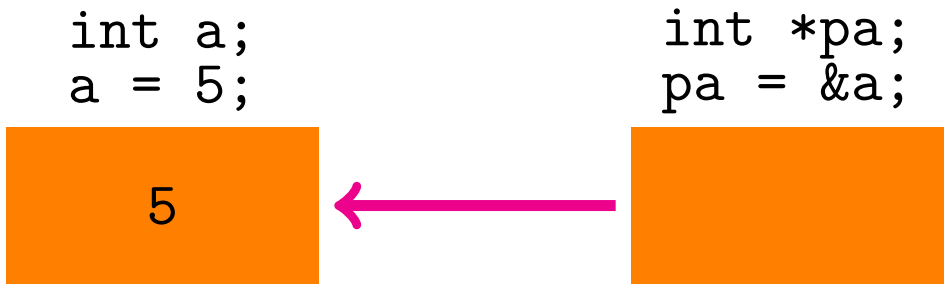

```
int *pa;
pa = &a;
```


int a;
a = 5;
int *pa;
pa = &a;
5



Task

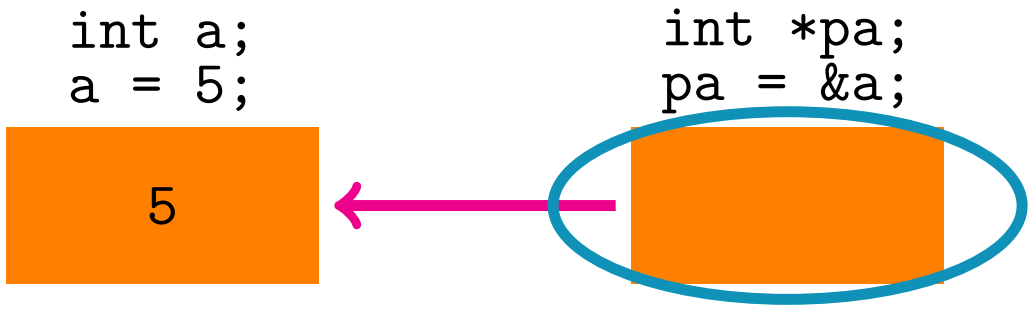
Set the value of a to 10 using pa.



Task

Set the value of a to 10 using pa.

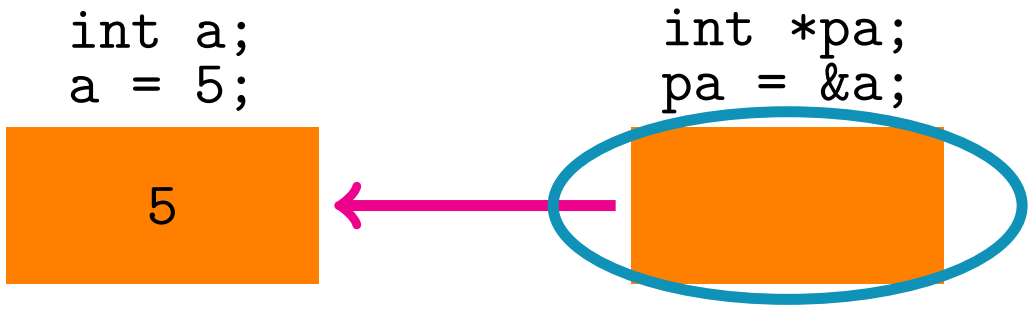
pa



Task

Set the value of a to 10 using pa.

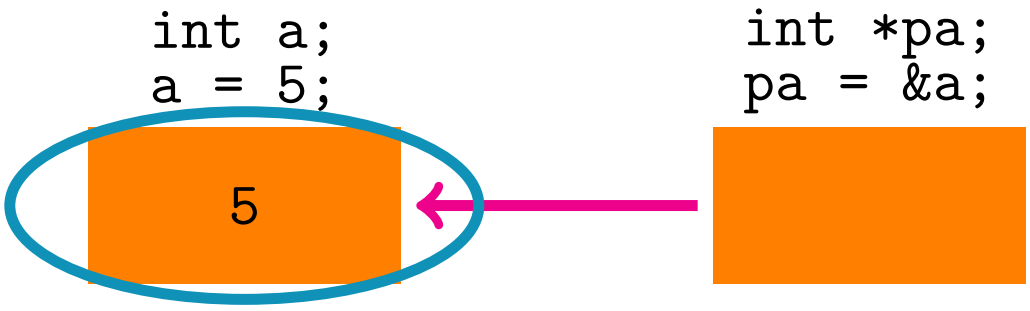
pa



Task

Set the value of a to 10 using pa.

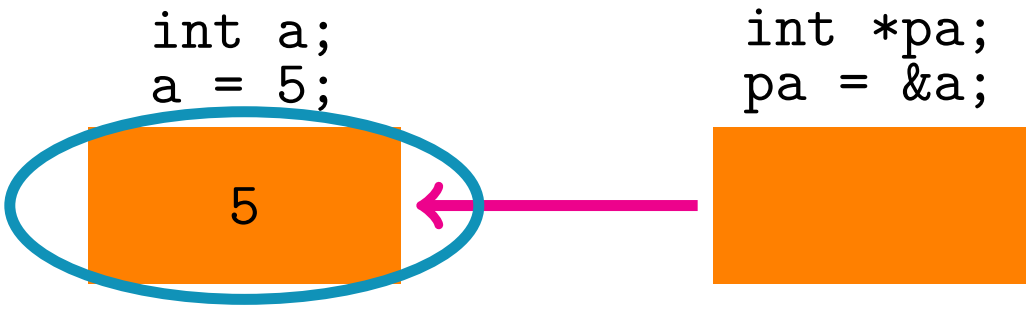
```
*pa
```



Task

Set the value of a to 10 using pa.

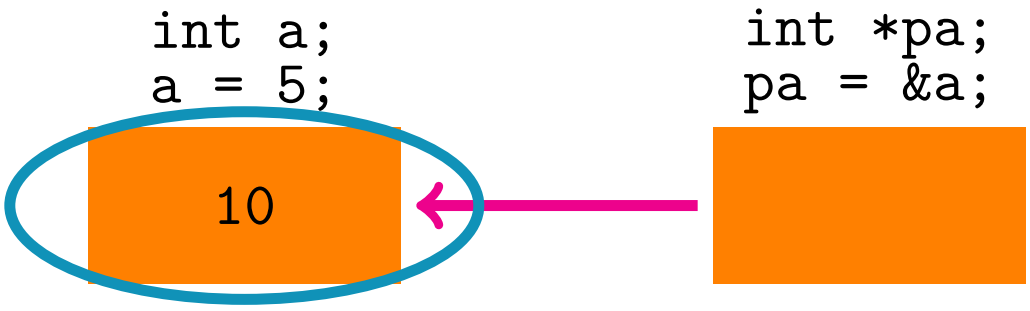
```
*pa
```



Task

Set the value of a to 10 using pa.

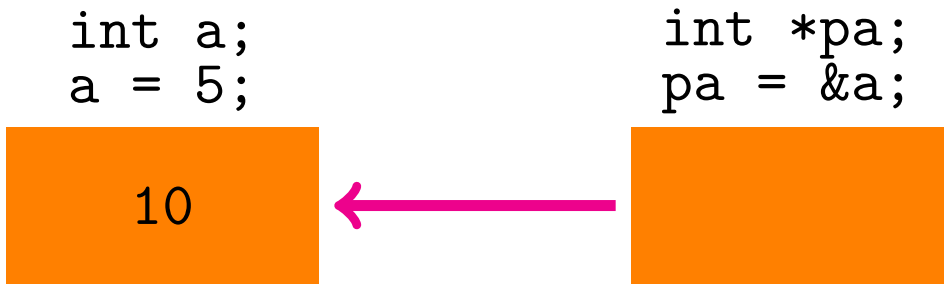
```
*pa = 10;
```

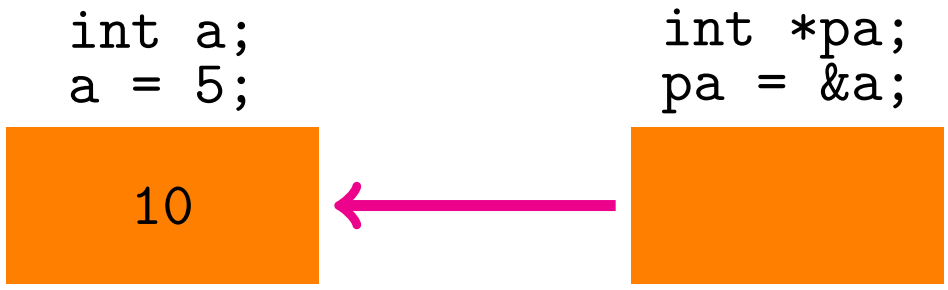


Task

Set the value of a to 10 using pa.

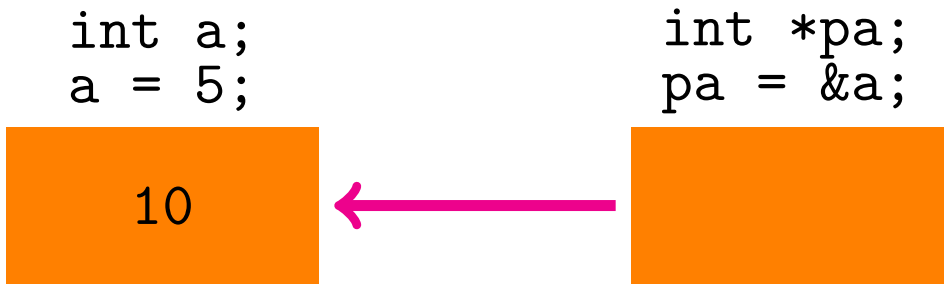
```
*pa = 10;
```


int a;
a = 5;
int *pa;
pa = &a;
10



Task

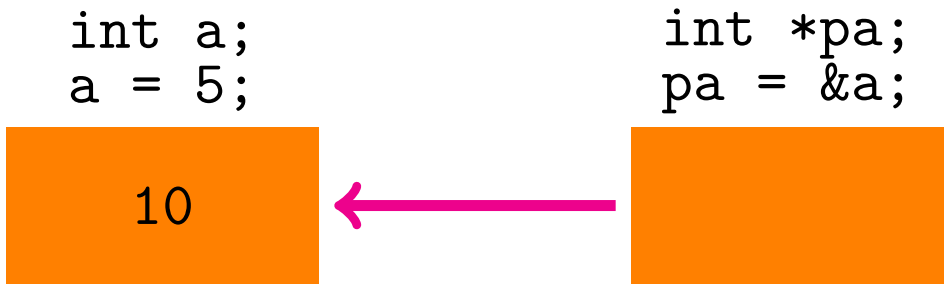
Create a pointer to `pa`.



Task

Create a pointer to pa.

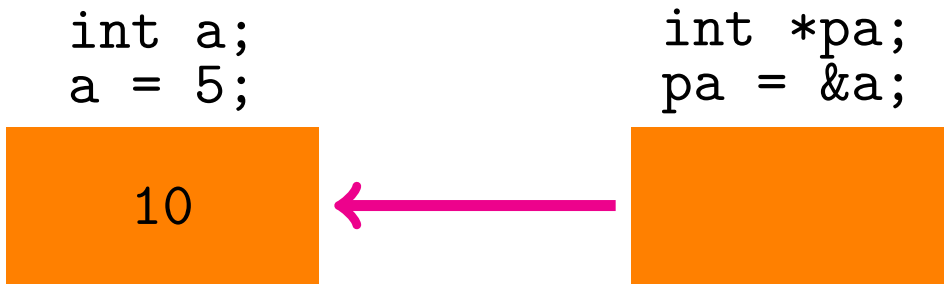
Data Type	Pointer to Data Type



Task

Create a pointer to `pa`.

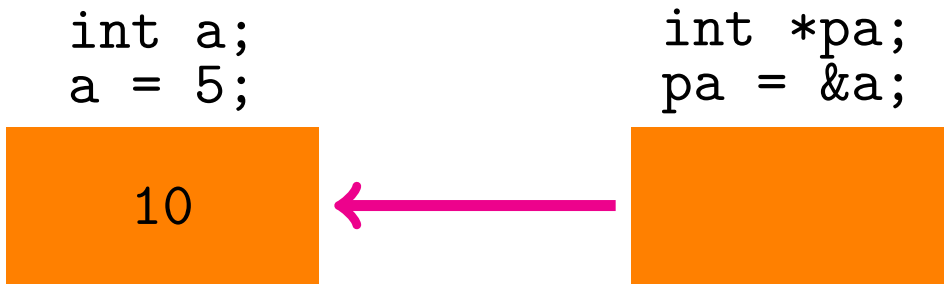
Data Type	Pointer to Data Type
`int`	`int *`



Task

Create a pointer to `pa`.

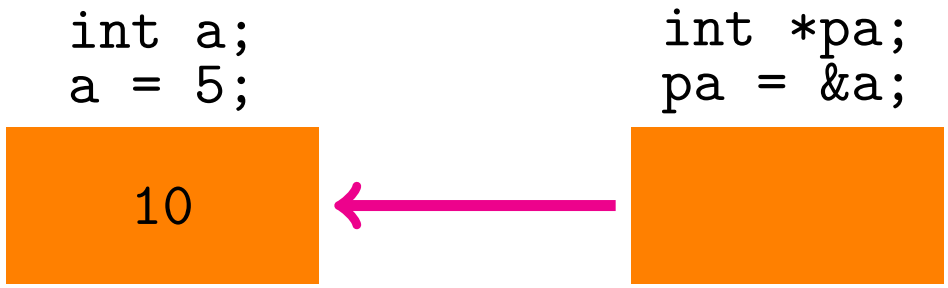
Data Type	Pointer to Data Type
`int`	`int *`
Integer	



Task

Create a pointer to `pa`.

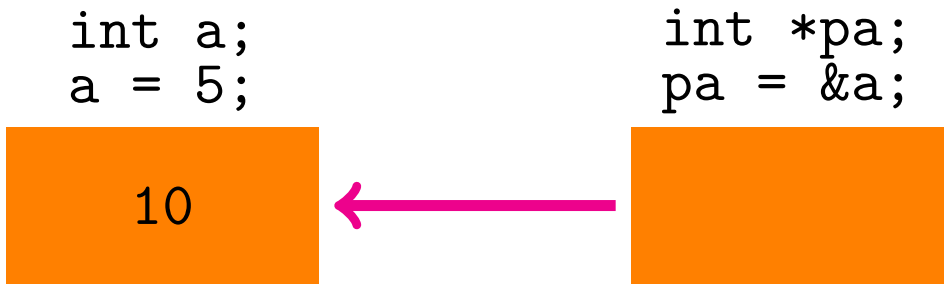
Data Type	Pointer to Data Type
`int`	`int *`
Integer	*Pointer to integer*



Task

Create a pointer to `pa`.

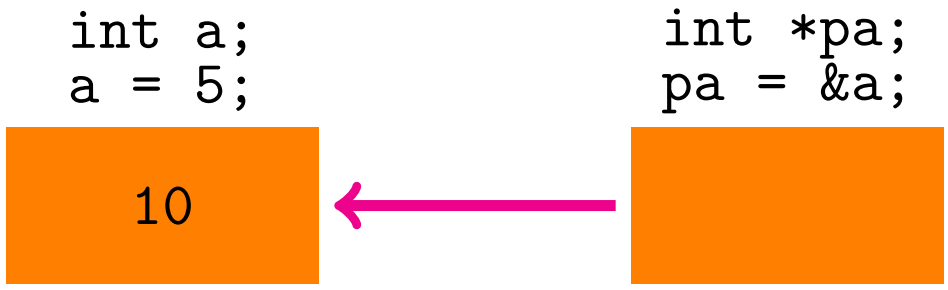
Data Type	Pointer to Data Type
`int`	`int *`
Integer	*Pointer to integer*
`int *`	



Task

Create a pointer to `pa`.

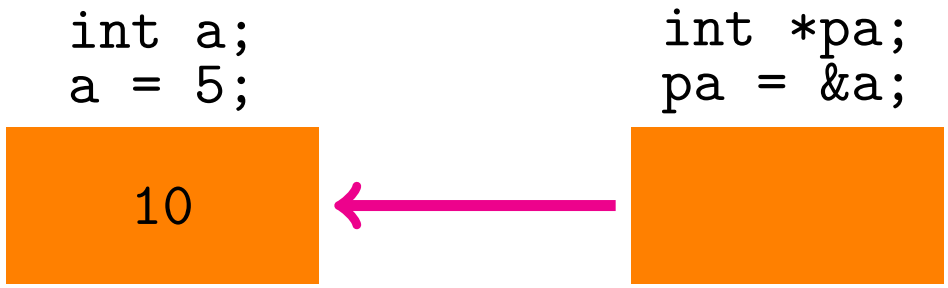
Data Type	Pointer to Data Type
`int`	`int *`
Integer	*Pointer to integer*
`int *`	
Pointer to integer	



Task

Create a pointer to `pa`.

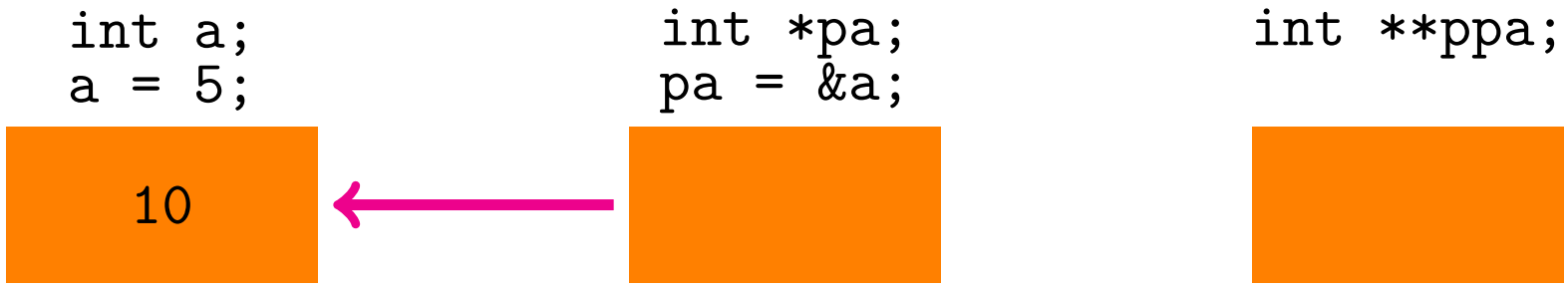
Data Type	Pointer to Data Type
`int`	`int *`
Integer	*Pointer to integer*
`int *`	
Pointer to integer	Pointer to pointer to integer



Task

Create a pointer to `pa`.

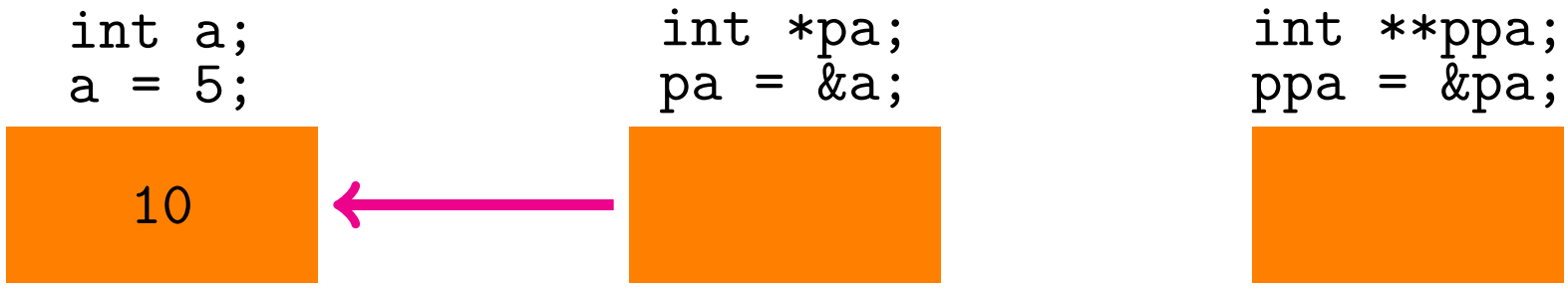
Data Type	Pointer to Data Type
`int`	`int *`
Integer	*Pointer to integer*
`int *`	`int **`
Pointer to integer	Pointer to pointer to integer



Task

Create a pointer to `pa`.

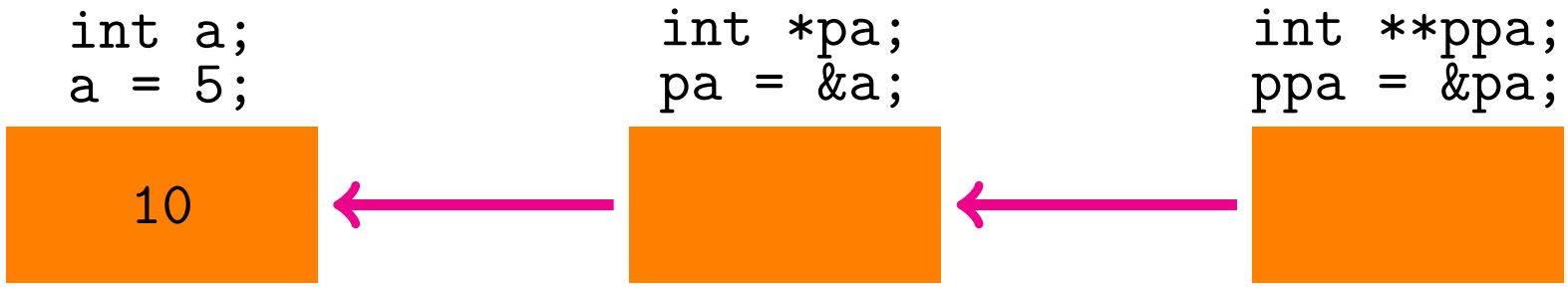
Data Type	Pointer to Data Type
`int`	`int *`
Integer	*Pointer to integer*
`int *`	`int **`
Pointer to integer	Pointer to pointer to integer



Task

Create a pointer to `pa`.

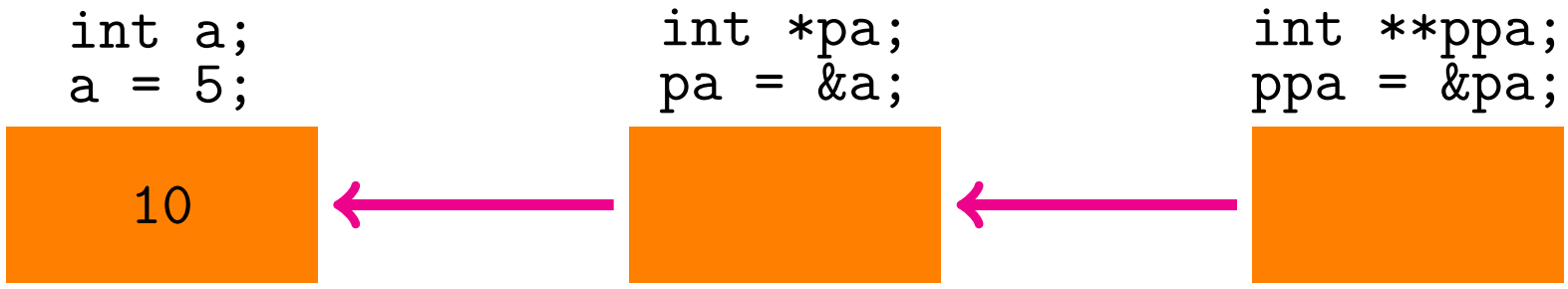
Data Type	Pointer to Data Type
`int`	`int *`
Integer	*Pointer to integer*
`int *`	`int **`
Pointer to integer	Pointer to pointer to integer

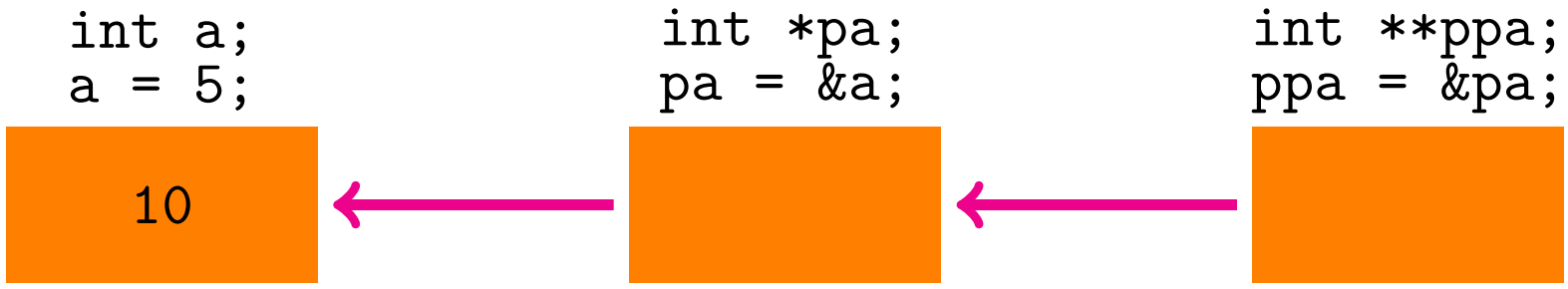


Task

Create a pointer to `pa`.

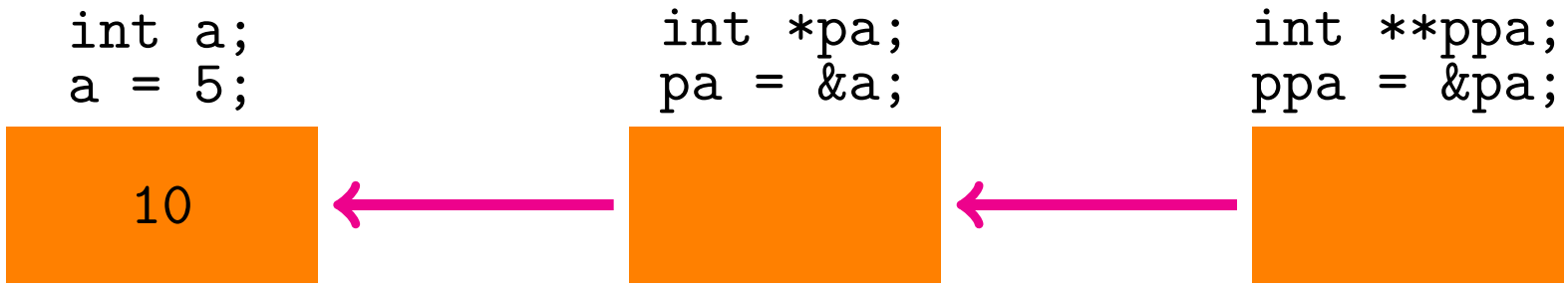
Data Type	Pointer to Data Type
`int`	`int *`
Integer	*Pointer to integer*
`int *`	`int **`
Pointer to integer	Pointer to pointer to integer


int a;
a = 5;
int *pa;
pa = &a;
int **ppa;
ppa = &pa;
10



Task

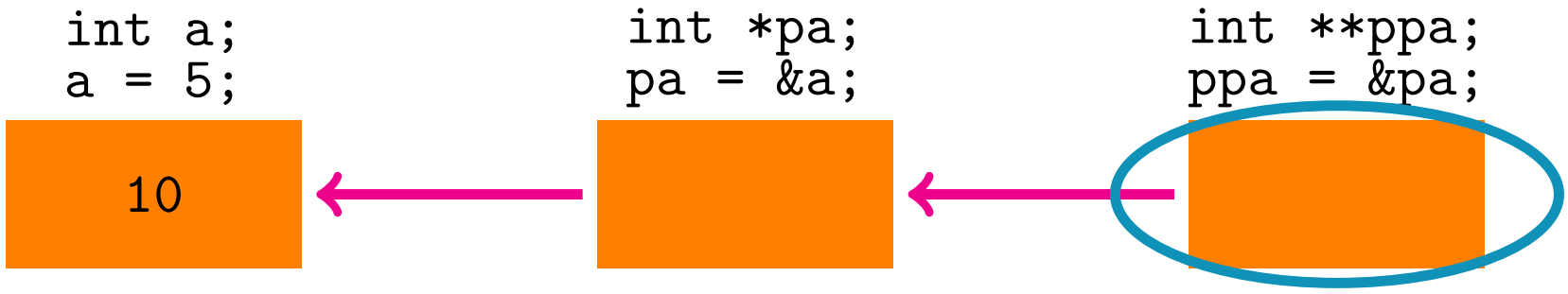
Set the value of a to 100 using ppa.



Task

Set the value of a to 100 using ppa.

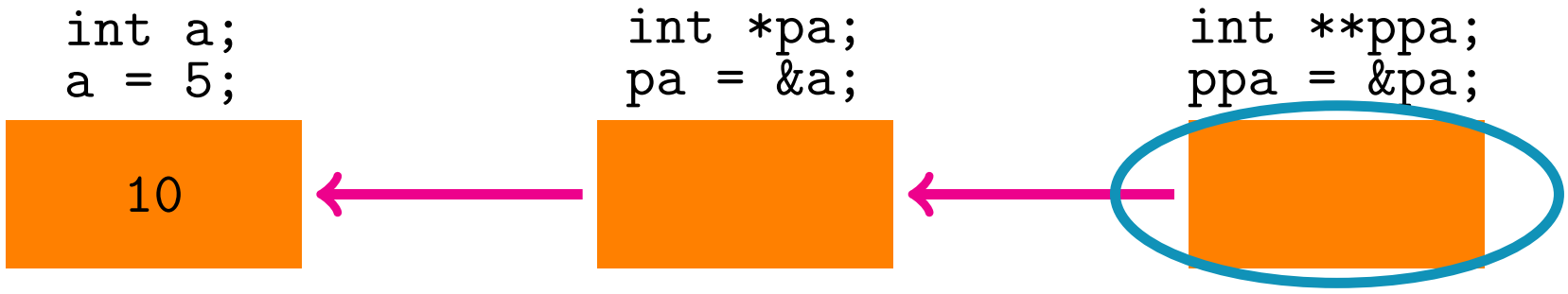
ppa



Task

Set the value of a to 100 using ppa.

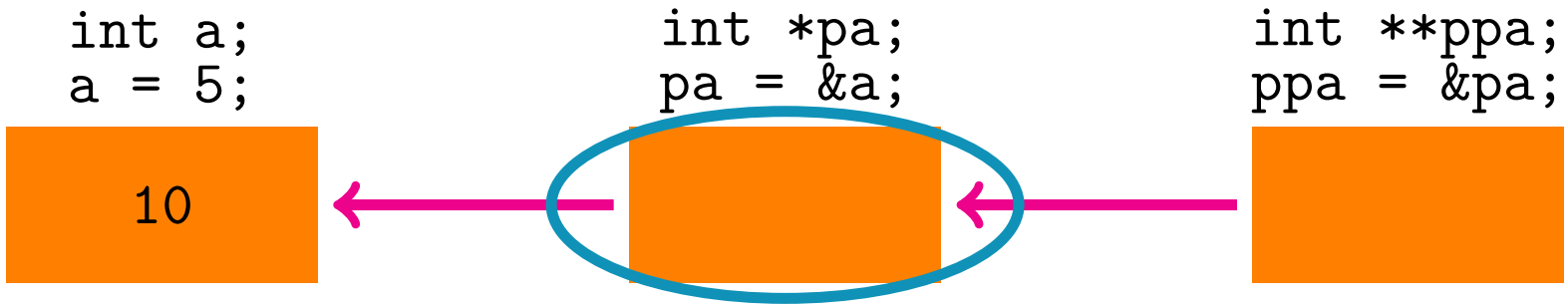
ppa



Task

Set the value of a to 100 using ppa.

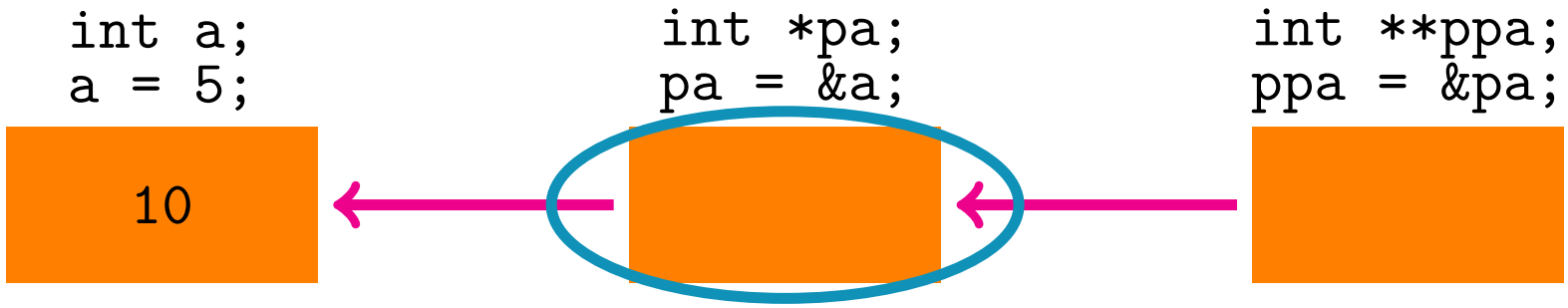
```
*ppa
```



Task

Set the value of a to 100 using ppa.

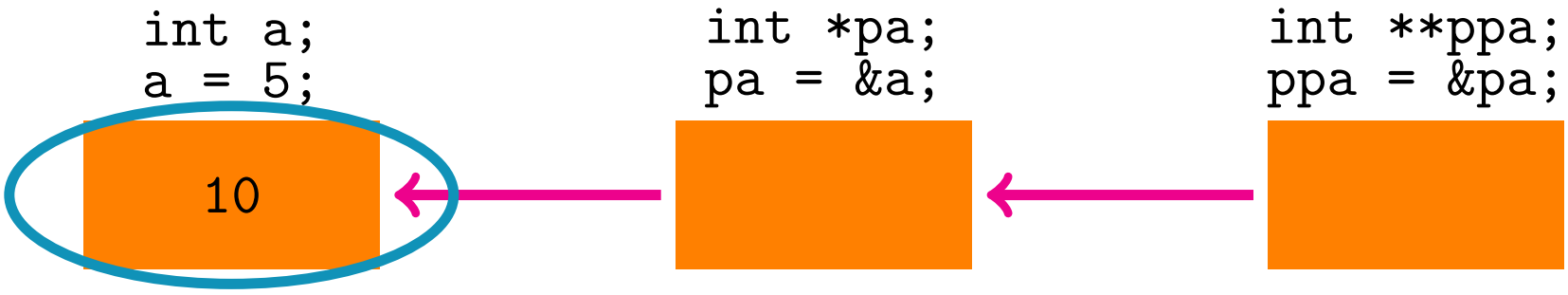
```
*ppa
```



Task

Set the value of a to 100 using ppa.

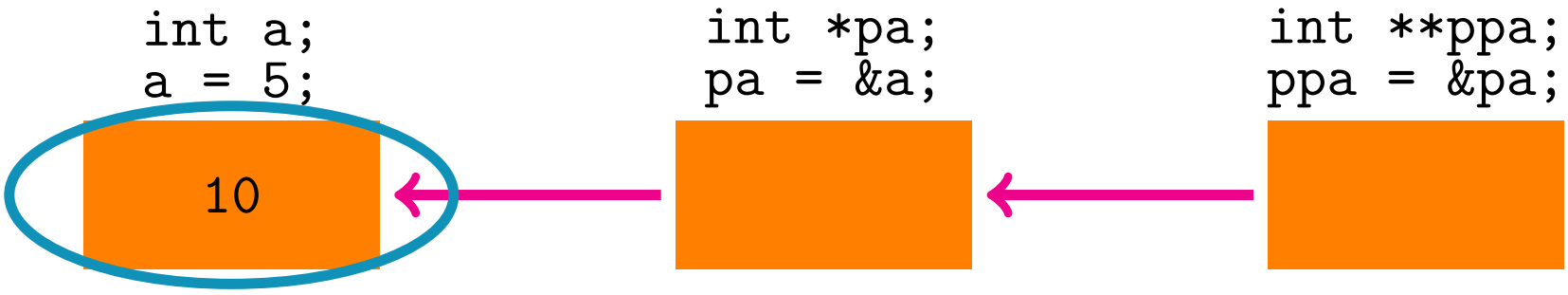
```
**ppa
```



Task

Set the value of a to 100 using ppa.

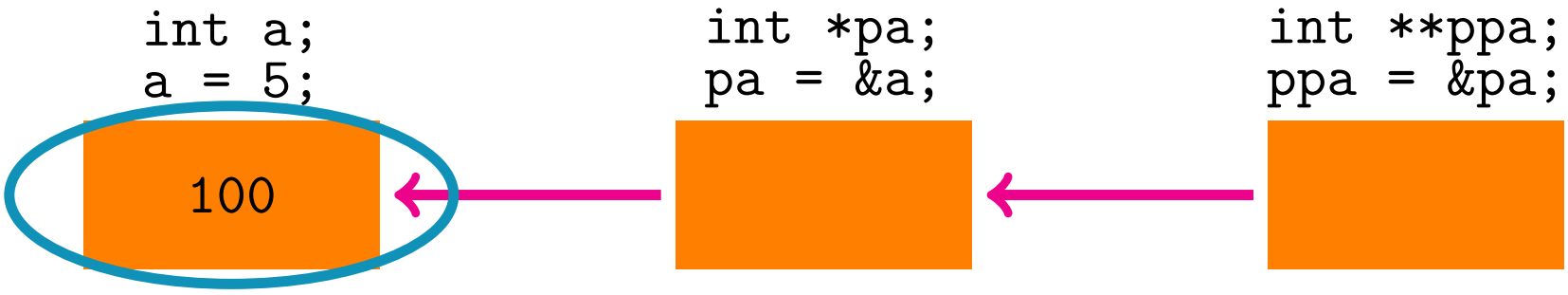
```
**ppa
```



Task

Set the value of a to 100 using ppa.

```
**ppa = 100;
```



Task

Set the value of a to 100 using ppa.

```
**ppa = 100;
```

Rule to create pointers

Rule to create pointers

(data type)

Rule to create pointers

(data type) (data type)*

Rule to create pointers

(data type)	(data type)*
int	

Rule to create pointers

(data type)	(data type)*
int	int *

Rule to create pointers

(data type)	(data type)*
int	int *
int *	

Rule to create pointers

(data type)	(data type)*
int	int *
int *	int **

Rule to create pointers

(data type)	(data type)*
int	int *
int *	int **
int **	

Rule to create pointers

(data type)	(data type)*
int	int *
int *	int **
int **	int ***

Rule to create pointers

(data type)	(data type)*
int	int *
int *	int **
int **	int ***
float	

Rule to create pointers

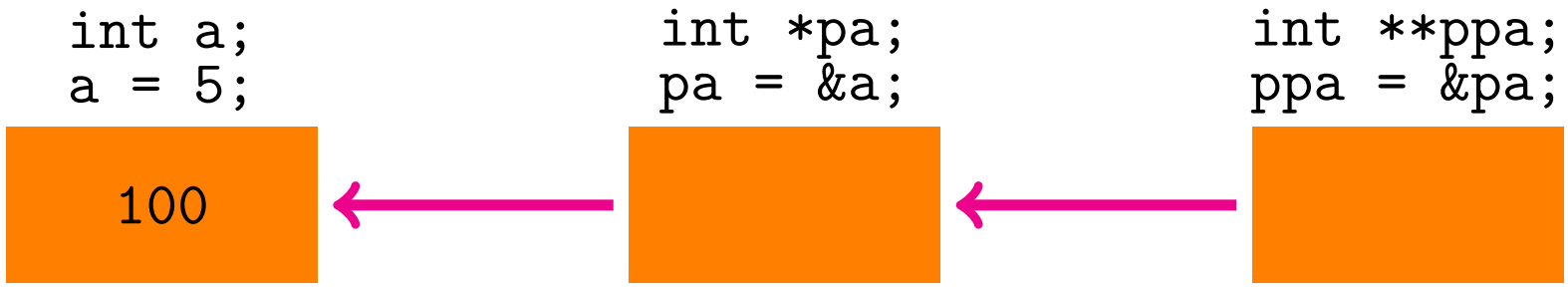
(data type)	(data type)*
int	int *
int *	int **
int **	int ***
float	float *

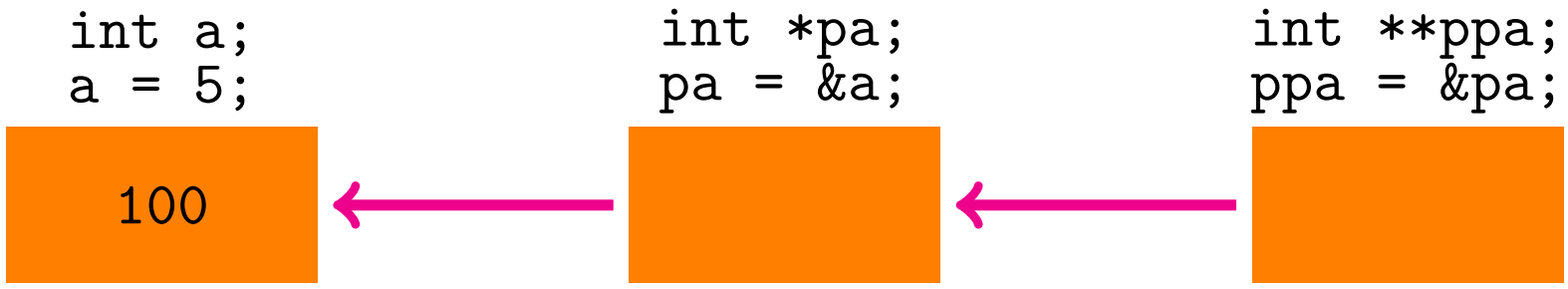
Rule to create pointers

(data type)	(data type)*
int	int *
int *	int **
int **	int ***
float	float *
char *	

Rule to create pointers

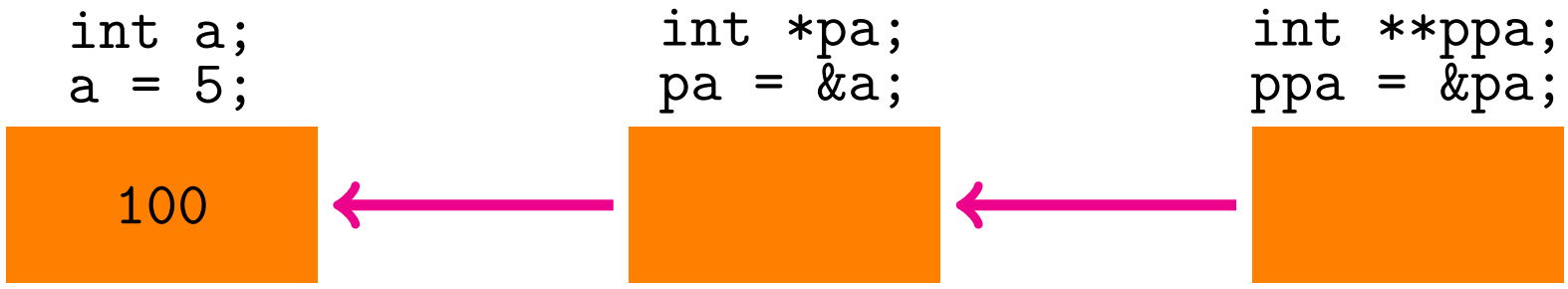
(data type)	(data type)*
int	int *
int *	int **
int **	int ***
float	float *
char *	char **


int a;
a = 5;
int *pa;
pa = &a;
int **ppa;
ppa = &pa;
100



Code segment

```
int *b;
b = pa;
printf("%d", *b);
```

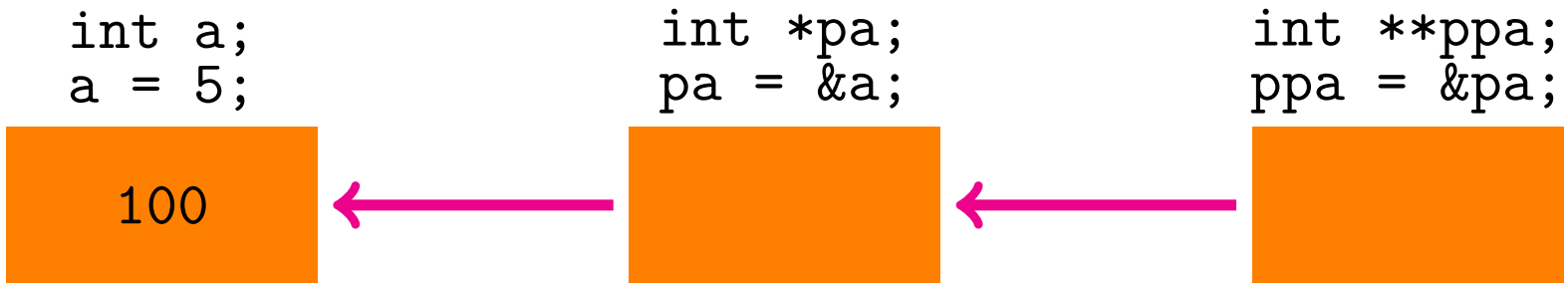


Code segment

```
int *b;
b = pa;
printf("%d", *b);
```

Question

Will it compile?

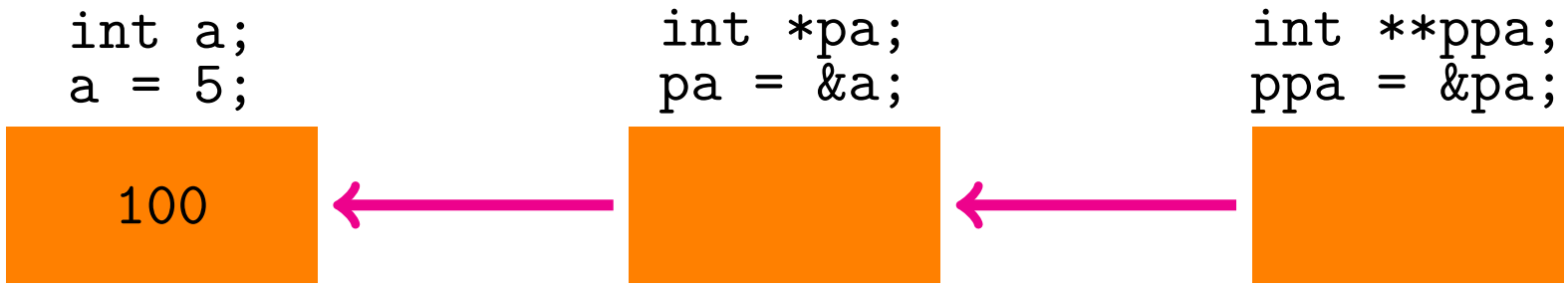


Code segment

```
int *b;
b = pa;
printf("%d", *b);
```

Question

Will it compile? ✓



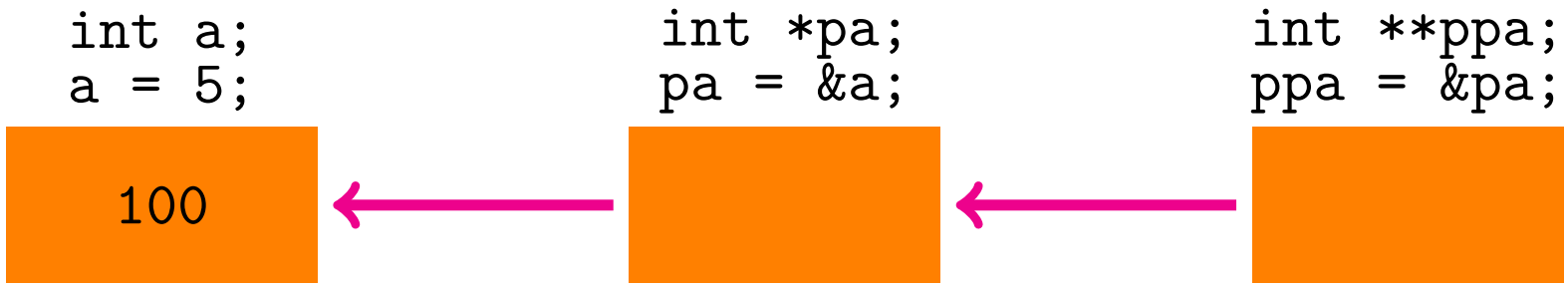
Code segment

```
int *b;
b = pa;
printf("%d", *b);
```

Question

Will it compile? ✓

What is the output?



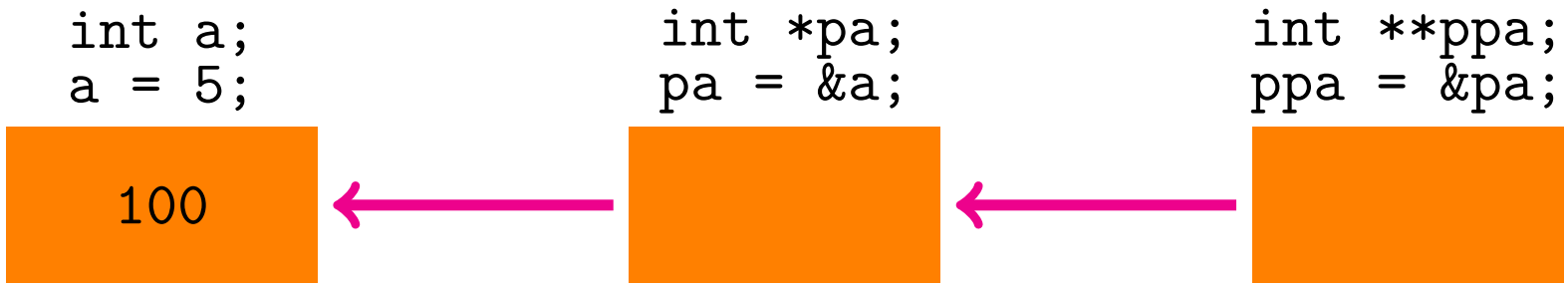
Code segment

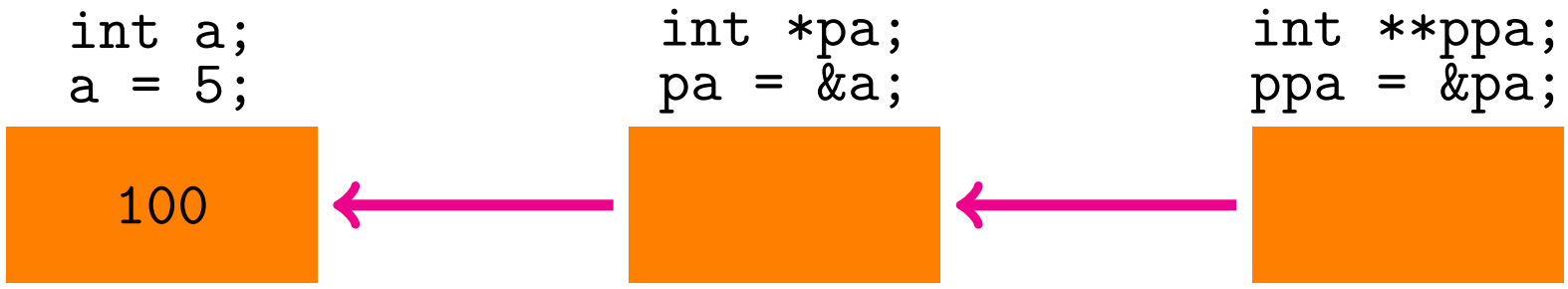
```
int *b;
b = pa;
printf("%d", *b);
```

Question

Will it compile? ✓

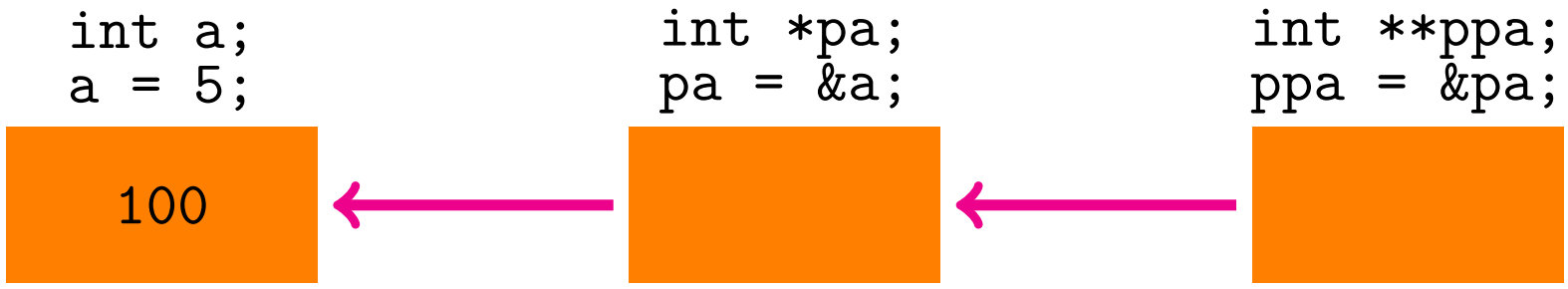
What is the output? → 100


int a;
a = 5;
int *pa;
pa = &a;
int **ppa;
ppa = &pa;
100



Code segment

```
int *b;
pa = b;
printf("%d", *b);
```

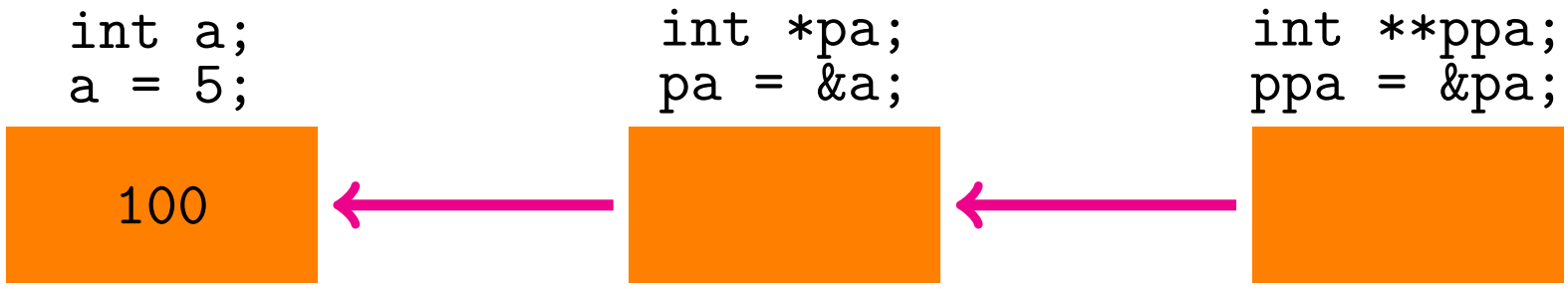


Code segment

```
int *b;
pa = b;
printf("%d", *b);
```

Question

Will it compile?

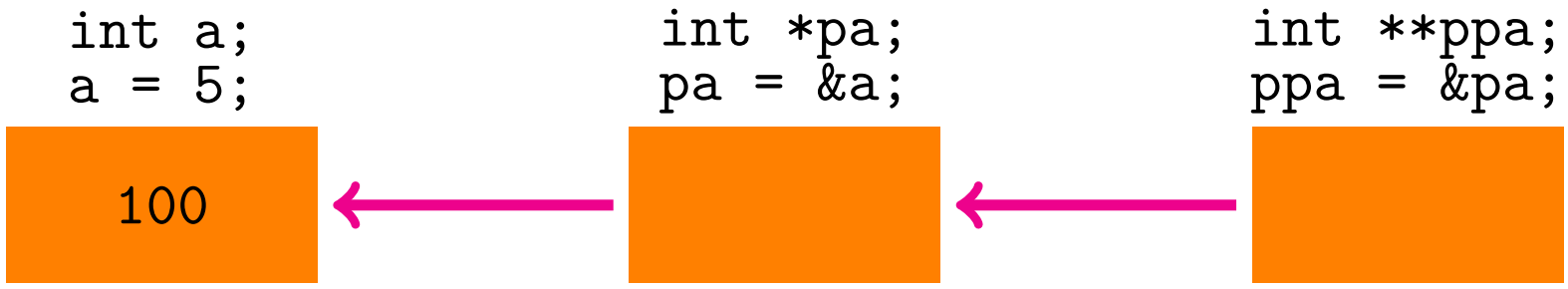


Code segment

```
int *b;
pa = b;
printf("%d", *b);
```

Question

Will it compile? ✓



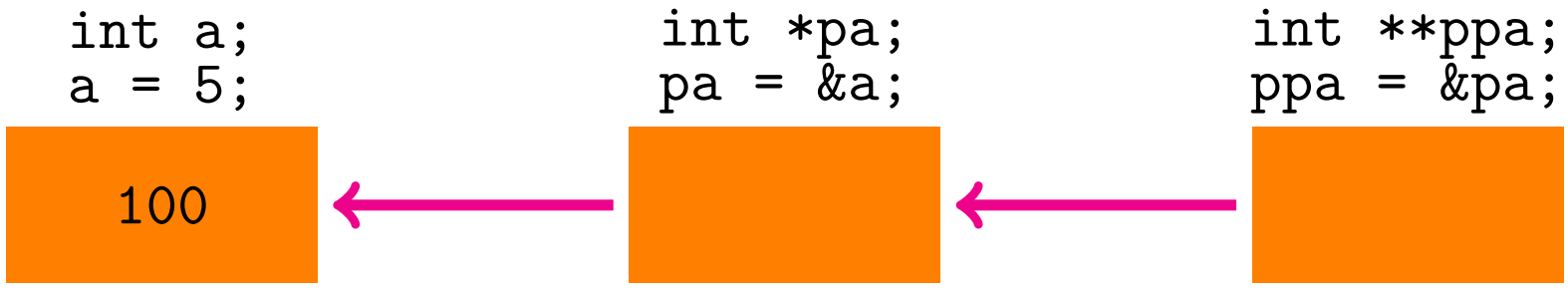
Code segment

```
int *b;
pa = b;
printf("%d", *b);
```

Question

Will it compile? ✓

What is the output?



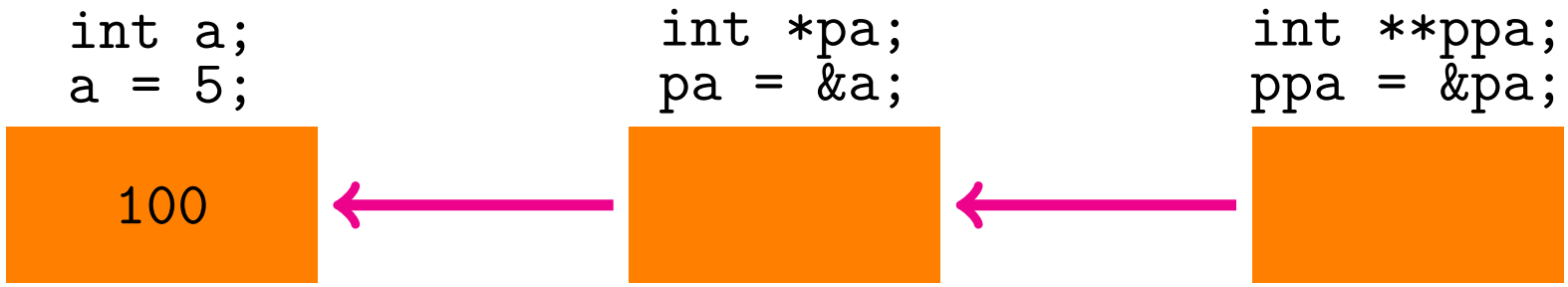
Code segment

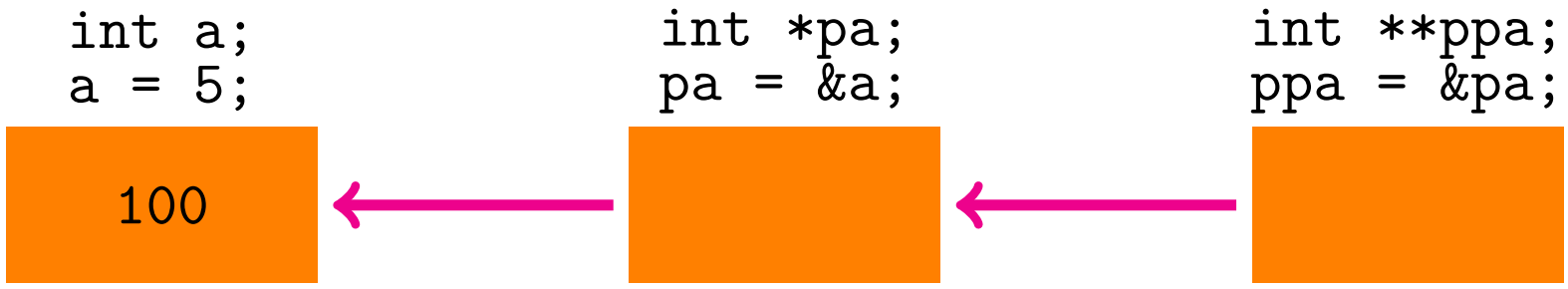
```
int *b;
pa = b;
printf("%d", *b);
```

Question

Will it compile? ✓

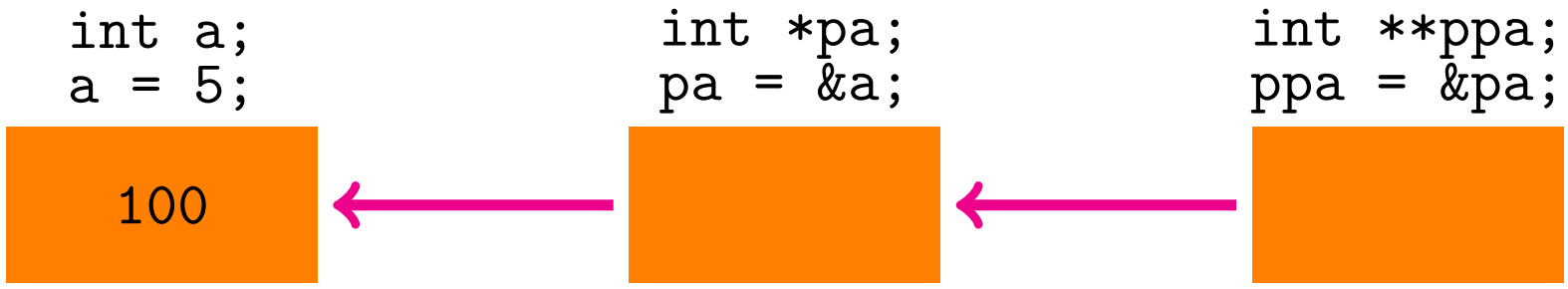
What is the output? → Garbage, perhaps even a segmentation fault.


int a;
a = 5;
int *pa;
pa = &a;
int **ppa;
ppa = &pa;
100



Code segment

```
ppa = pa;
printf("%d", *ppa);
```

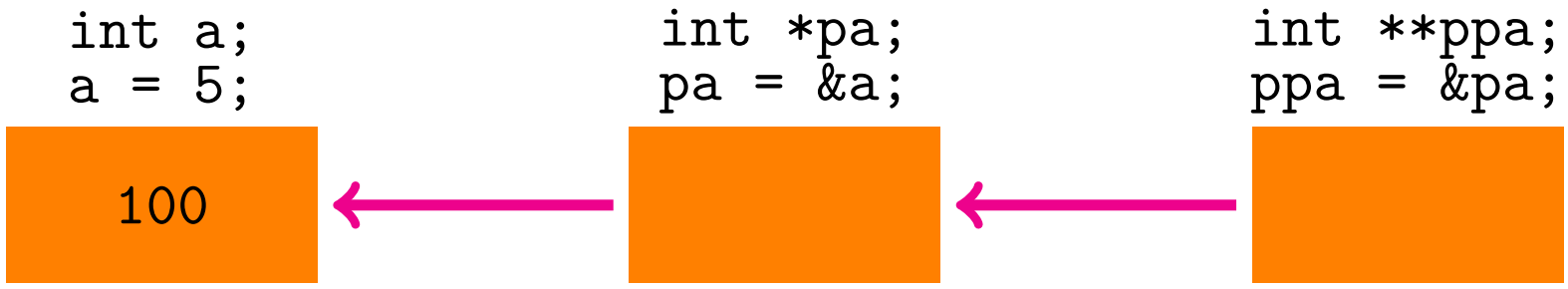


Code segment

```
ppa = pa;
printf("%d", *ppa);
```

Question

Will it compile?

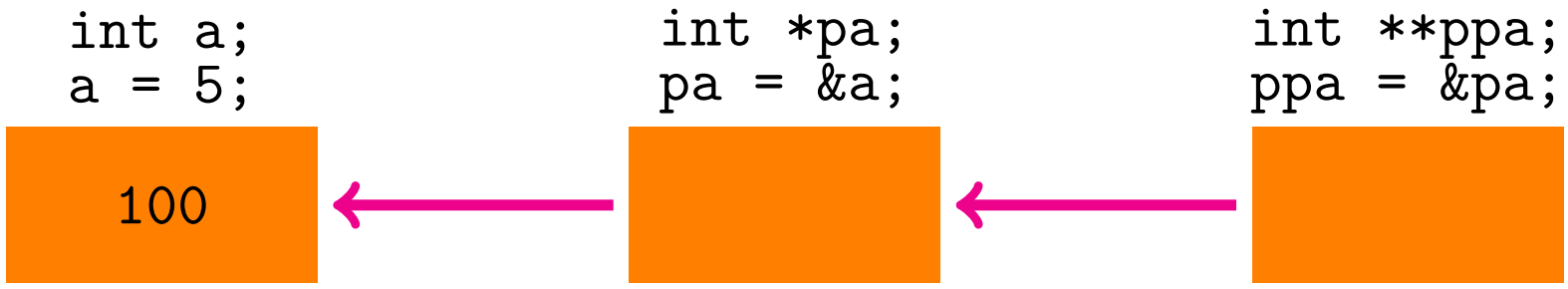


Code segment

```
ppa = pa;
printf("%d", *ppa);
```

Question

Will it compile? ✓(unfortunately)



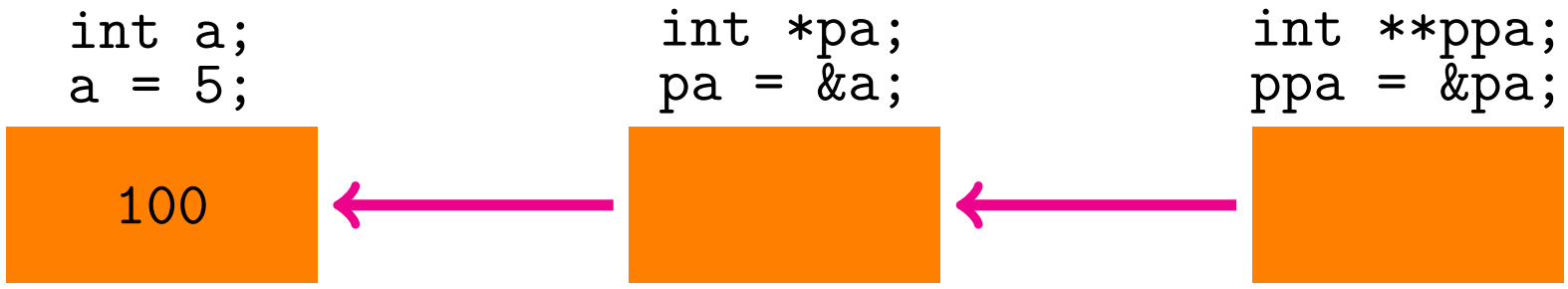
Code segment

```
ppa = pa;
printf("%d", *ppa);
```

Question

Will it compile? ✓(unfortunately)

What is the output?



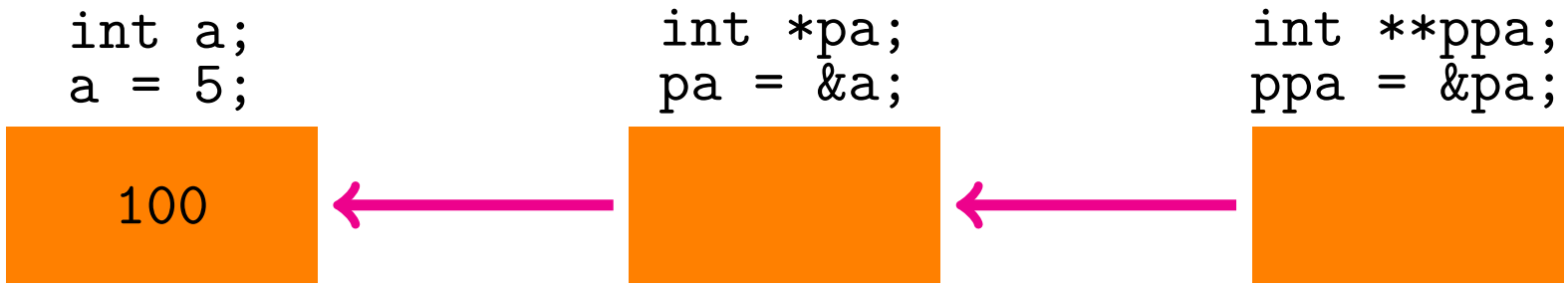
Code segment

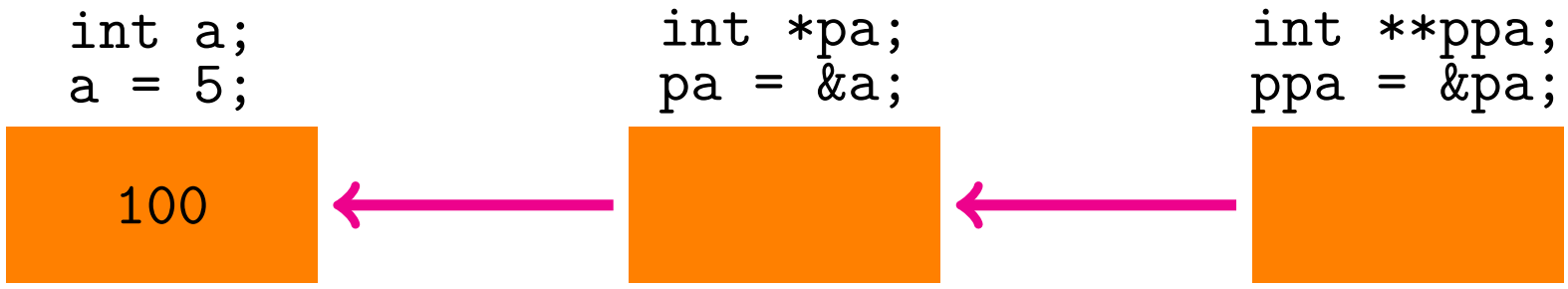
```
ppa = pa;
printf("%d", *ppa);
```

Question

Will it compile? ✓(unfortunately)

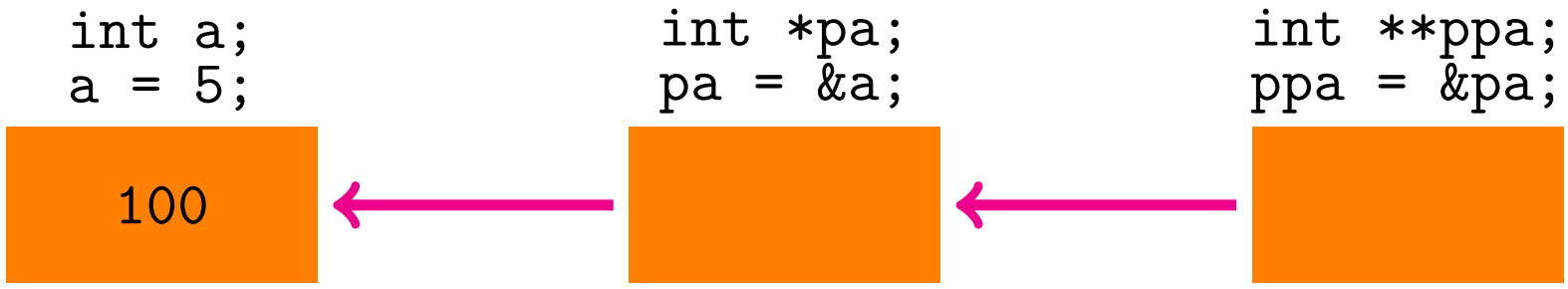
What is the output? $\rightarrow$ Unpredictable.


int a;
a = 5;
int *pa;
pa = &a;
int **ppa;
ppa = &pa;
100



Code segment

```
pa = ppa;
printf("%d", *pa);
```

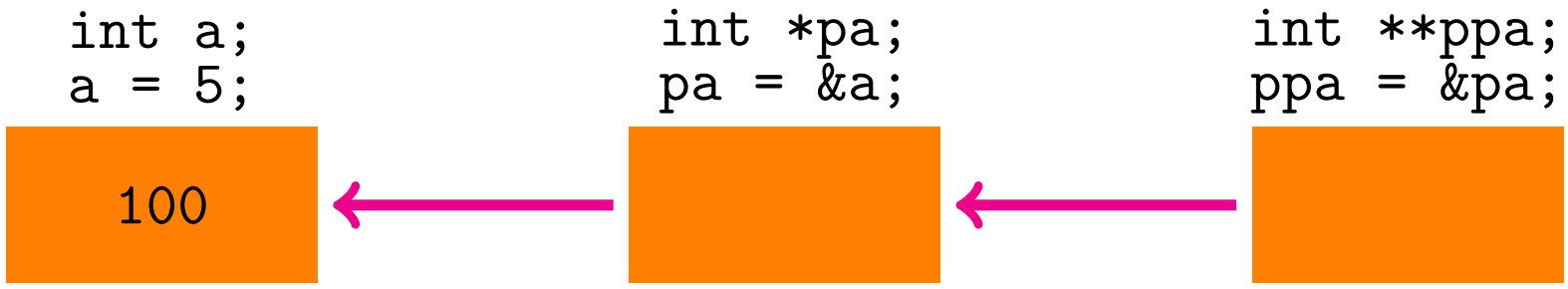


Code segment

```
pa = ppa;
printf("%d", *pa);
```

Question

Will it compile?

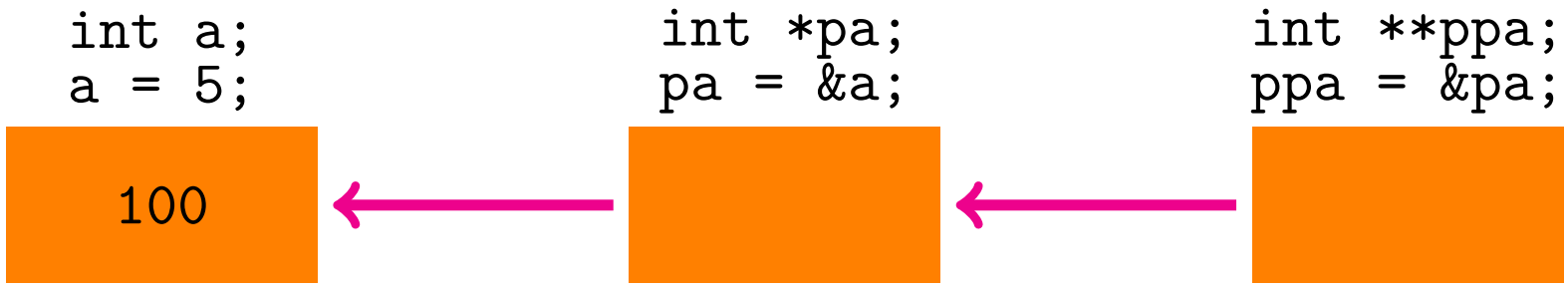


Code segment

```
pa = ppa;
printf("%d", *pa);
```

Question

Will it compile? ✓(unfortunately)



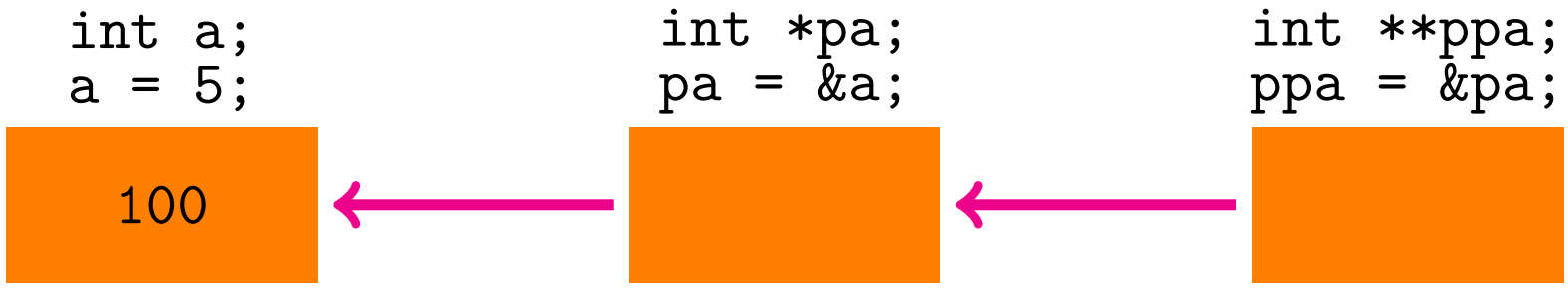
Code segment

```
pa = ppa;
printf("%d", *pa);
```

Question

Will it compile? ✓(unfortunately)

What is the output?



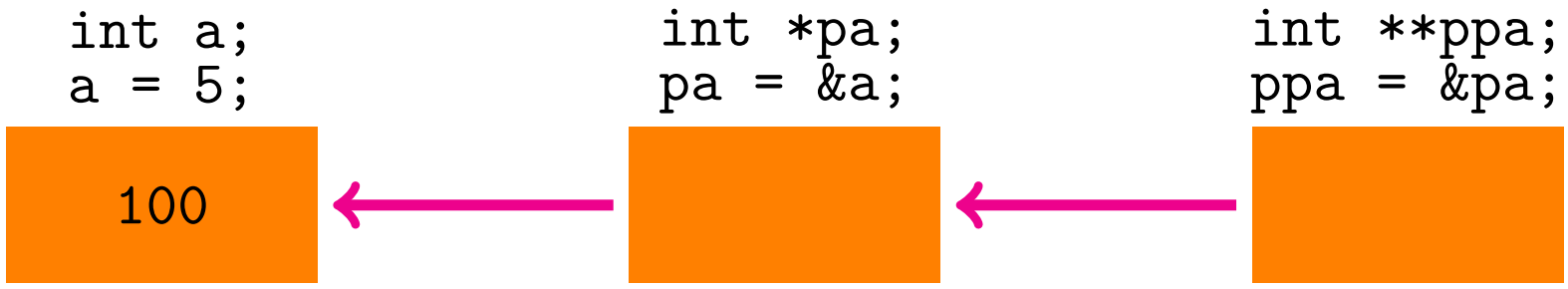
Code segment

```
pa = ppa;
printf("%d", *pa);
```

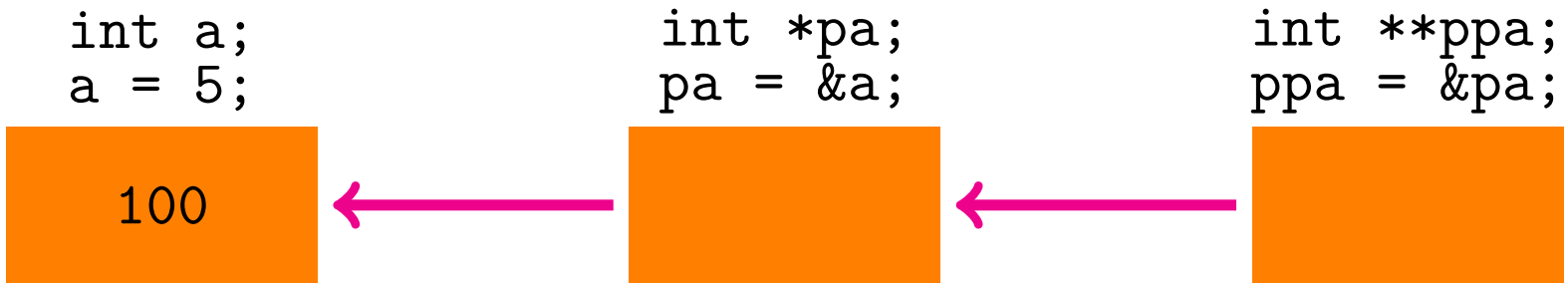
Question

Will it compile? ✓(unfortunately)

What is the output? $\rightarrow$ Unpredictable.

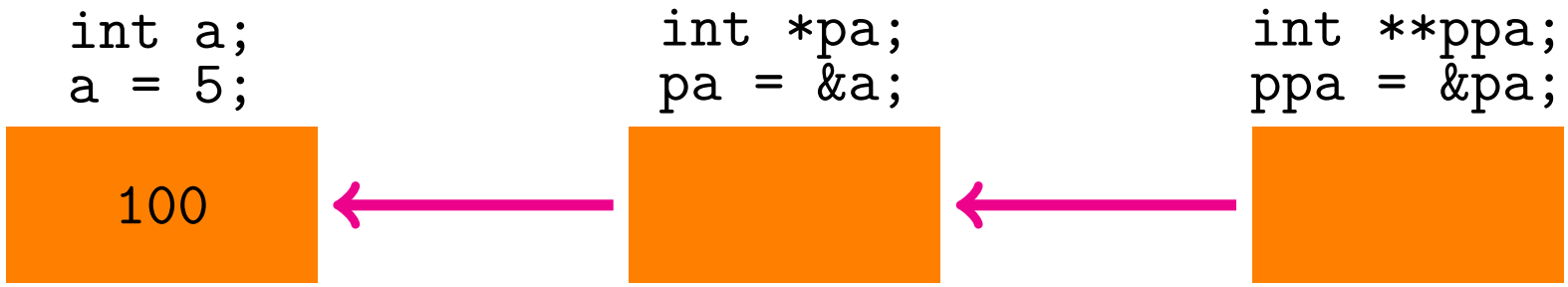


Comments



Comments

Onus is on the programmer to do the right thing.



Comments

Onus is on the programmer to do the right thing.

Do correct pointer operations to avoid unpredictable program behaviour.