

Consider the following code:

```
typedef struct Point {
    int x;
    int y;
} Point;

void changeX1 (Point pt) {
    pt.x = 11;
}
void changeX2 (Point *pt) {
    pt.x = 22;
}
void changeX3 (Point *pt) {
    pt->x = 33;
}
void changeX4 (Point *pt) {
    (*pt).x = 44;
}
void changeX5 (Point *pt) {
    pt = 55;
}

int main() {
    // main body - our code goes here
}
```

Fill in the blank with the printed value if we replace the "main body" with each of the following, and included only the version of change used in the main body.

If the value printed is uncertain, enter "garbage". If this program would cause a compile-time error or the behavior is uncertain, enter "n/a".

Q1.1:

```
Point my_pt = {1, 2};
changeX1(my_pt);
printf("%d\n", my_pt.x);
```

1

? ✓ 100%

Q1.2:

```
Point my_pt;
changeX1(my_pt);
printf("%d\n", my_pt.x);
```

garbage

? ✓ 100%

Q1.3:

```
Point my_pt;
changeX2(&my_pt);
printf("%d\n", my_pt.x);
```

n/a

? ✓ 100%

Q1.4:

```
Point *my_pt = malloc(sizeof(Point));
changeX3(my_pt);
printf("%d\n", my_pt->x);
```

33

? ✓ 100%

Q1.5:

Homework 2

Assessment  
overview

Total points: 65/100

Score: 65%

Question

Value: 15

History: 15

Awarded points: 15/15

Report an error in this  
question

Previous question

Next question

Attached files

No attached files

Attach a file

Attach text

```
Point *my_pt = calloc(1, sizeof(Point));
changeX5(my_pt);
printf("%d\n", my_pt->x);
```

0

? ✓ 100%

Q1.6:

```
Point my_pt;
changeX4(&my_pt);
printf("%d\n", my_pt.x);
```

44

? ✓ 100%

Try a new variant

Correct answer

Q1.1:

```
Point my_pt = {1, 2};
changeX1(my_pt);
printf("%d\n", my_pt.x);
```

1

Q1.2:

```
Point my_pt;
changeX1(my_pt);
printf("%d\n", my_pt.x);
```

garbage

Q1.3:

```
Point my_pt;
changeX2(&my_pt);
printf("%d\n", my_pt.x);
```

n/a

Q1.4:

```
Point *my_pt = malloc(sizeof(Point));
changeX3(my_pt);
printf("%d\n", my_pt->x);
```

33

Q1.5:

```
Point *my_pt = calloc(1, sizeof(Point));
changeX5(my_pt);
printf("%d\n", my_pt->x);
```

0

Q1.6:

```
Point my_pt;
changeX4(&my_pt);
printf("%d\n", my_pt.x);
```

44

Q1.1: C is pass-by-value. changeX gets a copy of the Point called pt and modifies it, which does nothing to change my\_pt. So my\_pt.x is still 1.

Q1.2: C is pass-by-value. changeX gets a copy of the Point called pt and modifies it, which does nothing to change my\_pt. Since my\_pt's fields were never initialized, its fields contain garbage.

Q1.3: The . operator to retrieve a field from a struct only works on a struct and not a struct pointer, so this code would not compile.

Q1.4: This code properly allocates space for the struct and passes a pointer by value. changeX3 dereferences the pointer and accesses the x field of the struct point using the -> operator, and changes x to 33.

Q1.5: C is pass by value. This code passes the value of the pointer to changeX5 which mutates its local copy, and does nothing to change my\_pt. Since the my\_pt was calloc-ed, all of its fields were initialized to 0 and remain equal to 0.

Q1.6: The pointer access in this code is equivalent to the one in the changeX3() example, just without using the shorthand -> notation. Using -> is pretty much always preferred, but it's important to know the two notations are equivalent.

Submitted answer 5 **correct: 100%**

Submitted at 2022-09-03 09:31:48 (PDT)



hide ^

Q1.1:

```
Point my_pt = {1, 2};
changeX1(my_pt);
printf("%d\n", my_pt.x);
```

1 **✓ 100%**

Q1.2:

```
Point my_pt;
changeX1(my_pt);
printf("%d\n", my_pt.x);
```

garbage **✓ 100%**

Q1.3:

```
Point my_pt;
changeX2(&my_pt);
printf("%d\n", my_pt.x);
```

n/a **✓ 100%**

Q1.4:

```
Point *my_pt = malloc(sizeof(Point));
changeX3(my_pt);
printf("%d\n", my_pt->x);
```

33 **✓ 100%**

Q1.5:

```
Point *my_pt = calloc(1, sizeof(Point));
changeX5(my_pt);
printf("%d\n", my_pt->x);
```

0 **✓ 100%**

Q1.6:

```
Point my_pt;
changeX4(&my_pt);
printf("%d\n", my_pt.x);
```

Submitted answer 4

partially correct: 83%

Submitted at 2022-09-03 09:31:07 (PDT)

i

show ▼

Submitted answer 3

partially correct: 83%

Submitted at 2022-09-03 09:29:55 (PDT)

i

show ▼

Show/hide older submissions ▼