

錬成問題

- 右に示すのは、三つの をもつ構造体の宣言である。ここで、`xyz` は と呼ばれ、その構造体に与える名前である。また、`a`、`b` はこの構造体の型をもつ実体であるオブジェクトである。複数の要素をもつという点で、構造体は配列と共通する点が多く、これらをまとめて 型と呼ぶ。

```
struct xyz {
    int    x;
    long   y;
    double z;
} a, b;
```

※次問以降、構造体 `xyz` は、本問で宣言されたものを意味する。

- 右に示すのは、構造体 `xyz` の型をもつオブジェクト `x` の である `x`、`y`、`z` に、それぞれ 1、2、3.14 を代入するプログラム部分である。

```
 = 1;
 = 2;
 = 3.14;
```

- `p` が構造体 `xyz` の型のオブジェクトを指している。右に示すのは、`p` が指すオブジェクトの である `x`、`y`、`z` に、それぞれ 1、2、3.14 を代入するプログラム部分である。

```
 = 1;
 = 2;
 = 3.14;
```

- 右に示すプログラム部分の実行結果を示せ。

<input type="text" value="(10)"/>
<input type="text" value="(11)"/>
<input type="text" value="(12)"/>

```
struct xyz x = {3};
printf("%d\n", );
printf("%ld\n", );
printf("%.1f\n", );
```

- 右に示すプログラム部分の実行結果を示せ。

<input type="text" value="(13)"/>
<input type="text" value="(14)"/>
<input type="text" value="(15)"/>

```
struct xyz x, y = {3};
x = y;
printf("%d\n", );
printf("%ld\n", );
printf("%.1f\n", );
```

- 右に示すのは、`p` が指す構造体 `xyz` の型のオブジェクトの である `x`、`y`、`z` の全てに 0 を代入する関数である。

```
void set0(struct xyz *p)
{
     = 0;
     = 0;
     = 0;
}
```

- 右に示すのは、 である `x`、`y`、`z` が全て 0 である構造体 `xyz` (の値) をそっくり返す関数である。

```
 xyz0(void)
{
     t =
        ;
    return ();
}
```

■ 右に示すのは、三つの (1) x, y, zをもつ構造体に XYZ という型名を与える宣言である。

※次問以降、XYZ型は、本問で宣言されたものを意味する。

```
(23) struct {
    int x;
    long y;
    double z;
} (24);
```

■ 右に示すのは、xが指すXYZ型オブジェクトの (1) である x, y, zの全てに0を代入する関数である。

```
void set0((25) x)
{
    (26) = 0;
    (27) = 0;
    (28) = 0;
}
```

■ 右に示すのは、(1) である x, y, zが全て0である XYZ 型構造体 (の値) をそっくり返す関数である。

```
(29) xyz0(void)
{
    (30) t = (31);

    return ((32));
}
```

■ (33) が異なれば、同一の有効範囲内においても、異なるものに対して同じ綴りの識別子を与えることができる。その (33) は4種類あり、それぞれ、(34)、(35)、(36)、(37) である。