

Quiz: GCC Warnungen

Wie kann der Code in Teil 2 *semantisch sinnvoll* abgeändert werden, so dass die zugehörige Warnung (`-Wimplicit-fallthrough=`) verschwindet?



```
switch(y) {  
    case 1: answer = "A"; break;  
    case 0: answer = "B"; break;  
    default: answer = "C";  
}
```

```
switch(y) {  
    case 1: answer = "A";  
        /*FALLTHROUGH*/  
    case 0: answer = "B";  
        /*FALLTHROUGH*/  
    default: answer = "C"  
}
```

Quiz: GCC Warnungen

Wie kann der Code in Teil 3 *semantisch sinnvoll* abgeändert werden, so dass die zugehörige Warnung (`-Wparentheses`) verschwindet?

```
if((arg = 0)) { return "A"; }  
else { return "B"; }
```

```
if(arg == 0) { return "A"; }  
else { return "B"; }
```

```
if(!arg) { return "A"; }  
else { return "B"; }
```

Quiz: GCC Warning-Flags

Beantworten Sie für jeden Programmteil: Erfolgt die Ausgabe der zugehörigen Warnung standardmäßig durch GCC oder muss diese zunächst aktiviert werden? Wird die Warnung nicht standardmäßig angezeigt, geben Sie an, welches Warning-Flag spezifiziert werden kann, um deren Ausgabe zu aktivieren.

	GCC default	-Wall	-Wextra
Teil 1 und Teil 3 (-Wparentheses)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Teil 2 (-Wimplicit-fallthrough=)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Teil 4 (-Wuninitialized)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Quiz: printf Ausgabe sicherstellen

Welche printf Ausgaben bezüglich des Wertes des Pointers array werden trotz des aborts standardmäßig auf der Konsole ausgegeben? Die printf, stdout und fflush Man-Pages können hierbei hilfreich sein.

```
printf("array = %p\n", array);  
abort();
```

```
printf("array = %p", array);  
printf(""); // Flush previous printf  
abort();
```

```
printf("array = %p", array);  
fflush(stdout);  
abort();
```