

Prawidłowe formatowanie ułatwia tworzenie, sprawdzanie, poprawianie, tzw. konserwację i pielęgnowanie kodu. Warto jest przyjąć zasady edycji prawidłowego formatowania (wybór):

1. W jednej linii może być tylko jedna instrukcja (zakończona średnikiem) – nie dotyczy np. instrukcji będących parametrami wywołania jakiejś funkcji,
2. Każdy blok zaczyna się pojedynczym nawiasem { i kończy podobnym }, przy czym :
 - a. Nawiasy te znajdują się jako jedyne znaki w swoich liniach,
 - b. Nawias początkowy i końcowy danego bloku są w tej samej kolumnie,
 - c. Kod bloku jest wysunięty w prawo o jedną tabulację w stosunku do nawiasów.
3. Funkcje warunkowe i pętle wymagają wysunięcia kodu, który zawierają.

Przykładowy program:

```
#include <windows.h>

#define OPOZNIENIE 10000

unsigned long __stdcall watek(void *v)
{
    int i;

    while(1)
    {
        for(i=0;i<OPOZNIENIE;i++)
            ; // tutaj powinna byc instrukcja, chocby pusta -
            // w tym przypadku zamierzona

        for(i=0;i<OPOZNIENIE;i++);
        // takie podejscie najczesciej uznawane jest jako (przypadkowy) blad

        // petle i bloki

        for(i=0;i<OPOZNIENIE;i++)
        {
            s++;
        }

        for(j=0;j<OPOZNIENIE;j++)
            for(i=0;i<OPOZNIENIE;i++)
            {
                s++;
            }

        for(j=0;j<OPOZNIENIE;j++)
        {
            z++;

            for(i=0;i<OPOZNIENIE;i++)
            {
                s++;
            }
        }
    }
}
```

```
// przykładowa konstrukcja z instrukcjami warunkowymi
if(z>0)
    z++;
else
    if(c>0)
        c++;
    else
        z--;
```

```
// przykładowa konstrukcja z instrukcjami warunkowymi + bloki
if(z>0)
{
    z++;
    c++;
}
else
    if(c>0)
    {
        c++;
        z--;
    }
    else
        z--;
```

```
}
```

```
}
```