

Wprowadzenie do programowania w języku C

grupa RKr, wtorek 16:15-18:00

lista nr 4 (na pracownię 05.11.2019) (wersja 1)

Zadanie 1. [10p na pracowni lub 5p po pracowni]

uwaga: Makra muszą rozwijać się do wartości/wyrażeń, a nie do instrukcji, tak aby można było składać makra.

Musisz poradzić sobie bez „do { ... } while(0)”, zmiennych, pętli, etc. Pamiętaj o kompletnym nawiasowaniu.

Trudniejsze makra można budować poprzez złożenia i operacje na podstawowych, tj. mask, wipe, fire, flip, etc.

[2x1p | A] Napisz/zaprogramuj niżej wyspecyfikowane makra.

`__maskbit(x, b)` : pokaże bit b na jego pozycji | $__maskbit(1_3 \bar{1}_2 1_1 0_0, 2) = 0_3 \bar{1}_2 0_1 0_0$ ((num))
| $b_3 b_2 b_1 b_0 \rightarrow 0_3 b_2 0_1 0_0$ ((sym))

`__wipebit(x, b)` : zgasi bit b | $__wipebit(0_3 \bar{1}_2 1_1 0_0, 2) = 0_3 \bar{0}_2 1_1 0_0$ ((num))
| $b_3 b_2 b_1 b_0 \rightarrow b_3 0_2 b_1 b_0$ ((sym))

[2x1p | B] Napisz/zaprogramuj niżej wyspecyfikowane makra.

`__firebit(x, b)` : zapali bit b | $__firebit(\bar{0}_3 1_2 1_1 0_0, 3) = \bar{1}_3 1_2 1_1 0_0$ ((num))
| $b_3 b_2 b_1 b_0 \rightarrow 1_3 b_2 b_1 b_0$ ((sym))

`__flipbit(x, b)` : odwróci bit b | $__flipbit(0_3 1_2 \bar{1}_1 0_0, 1) = 0_3 1_2 0_1 0_0$ ((num))
| $b_3 b_2 b_1 b_0 \rightarrow b_3 b_2 \neg b_1 b_0$ ((sym))

[2x3p | C] Napisz/zaprogramuj niżej wyspecyfikowane makra.

`__copybit(x, b1, b2)` : skopiuje bit b1 w miejsce b2 | $__copybit(0_3 1_2 1_1 0_0, 2, 3) = 1_3 1_2 1_1 0_0$ ((num))
| $b_3 b_2 b_1 b_0 \rightarrow b_2 b_2 b_1 b_0$ ((sym))

`__swapbit(x, b1, b2)` : zamieni bity b1 i b2 | $__swapbit(0_3 1_2 1_1 0_0, 2, 3) = 1_3 0_2 1_1 0_0$ ((num))
| $b_3 b_2 b_1 b_0 \rightarrow b_2 b_3 b_1 b_0$ ((sym))

Zadanie 2. [10p] Napisz/zaprogramuj jedno z niżej wyspecyfikowanych makr.

`__liftbit(x, b, k)` : przełoży bit b o k miejsc w lewo | $__liftbit(0_3 1_2 \bar{1}_1 0_0, 1, 2) = \bar{1}_3 0_2 1_1 0_0$ ((num))
| $b_3 b_2 b_1 b_0 \rightarrow b_1 b_3 b_2 b_0$ ((sym))

`__dropbit(x, b, k)` : przełoży bit b o k pozycji w prawo | $__dropbit(\bar{1}_3 0_2 1_1 0_0, 3, 2) = 0_3 1_2 \bar{1}_1 0_0$ ((num))
| $b_3 b_2 b_1 b_0 \rightarrow b_2 b_1 b_3 b_0$ ((sym))

Alternatywnie, makra mogą przekładać bit z pozycji b na nową pozycję c, zamiast o k miejsc.

Wybór należy do Ciebie. W tym zadaniu przyda się mieszanie operacji bitowych i arytmetycznych.

Zadanie 3. [10p] Dostępne w serwisie SKOS.