

Systemy komputerowe

Lista zadań nr 2

Na zajęcia 9 – 12 marca 2020

W zadaniach odnoszących się do języka C wolno używać **wyłącznie** instrukcji przypisania, operatorów bitowych, dodawania i odejmowania, przesunięć bitowych, testów równości i nierówności (`==` i `!=`), zmiennych pomocniczych oraz stałych! Należy wytłumaczyć dlaczego rozwiązanie działa! Wszystkie liczby mają szerokość 32 bitów.

Zadanie 1. Czy poniższe wyrażenia zawsze obliczą się do prawdy dla dwóch dowolnych wartości zmiennych `x` i `y` typu `int32_t`? Jeśli nie to podaj wartości, które prowadzą do obliczenia fałszu.

1. `(x > 0) || (x - 1 < 0)`
2. `(x & 7) != 7 || (x << 29 < 0)`
3. `(x * x) >= 0`
4. `x < 0 || -x <= 0`
5. `x > 0 || -x >= 0`
6. `(x | -x) >> 31 == -1`
7. `((uint32_t)x) >> 3 == ((uint32_t)x)/8`
8. `x >> 3 == x/8`
9. `x + y == (uint32_t)y + (uint32_t)x`

Zadanie 2. Podaj wyrażenie zawierające wyłącznie zmienne `x` i `y`, którego wartością logiczną jest wynik porównania `x < y` dla liczb (a) bez znaku (b) ze znakiem.

Wskazówka: Spróbuj rozwiązać zadanie samodzielnie, a następnie przeczytaj §2.11 książki „Uczta programistów”.

Zadanie 3. Podaj fragment kodu, który oblicza funkcję:

$$abs(x) = \begin{cases} x & \text{dla } x \geq 0 \\ -x & \text{dla } x < 0 \end{cases}$$

Skorzystaj z następującej własności: jeśli `b` jest wartością logiczną, to wyrażenie `b ? x : y` można przetłumaczyć do `b * x + !b * y`.

Wskazówka: Spróbuj rozwiązać zadanie samodzielnie, a następnie przeczytaj §2.4 książki „Uczta programistów”.

Zadanie 4. Podaj fragment kodu, który oblicza funkcję:

$$sign(x) = \begin{cases} -1 & \text{dla } x < 0 \\ 0 & \text{dla } x = 0 \\ 1 & \text{dla } x > 0 \end{cases}$$

Wskazówka: Spróbuj rozwiązać zadanie samodzielnie, a następnie przeczytaj §2.7 książki „Uczta programistów”.

Zadanie 5. Uzupełnij ciało funkcji zadeklarowanej poniżej.

```
/* Jeśli suma x+y mieści się w typie int32_t (nie powoduje
   nadmiaru/niedomiaru) zwróć 1, w p.p. 0 */
int tadd_ok(int32_t x, int32_t y);
```

Zadanie 6. Uzupełnij ciało funkcji zadeklarowanej poniżej.

```
/* Jeśli x zawiera nieparzystą liczbę jedynek zwróć 1, w p.p. 0 */
int32_t odd_ones(uint32_t x);
```

Wskazówka: Spróbuj rozwiązać zadanie samodzielnie, a następnie przeczytaj §5.2 książki „Uczta programistów”.

Zadanie 7. Standard IEEE 754-2008¹ definiuje liczby zmiennopozycyjne o szerokości 16-bitów. Oblicz ręcznie $3.984375 \cdot 10^{-1} + 3.4375 \cdot 10^{-1} + 1.771 \cdot 10^3$ używając liczb w tym formacie. Zapisz wynik binarnie i dziesiętnie. Czy wynik się zmieni jeśli najpierw wykonamy drugie dodawanie?

UWAGA! Domyślną metodą zaokrąglania w obliczeniach zmiennoprzecinkowych jest *round-to-even*.

Zadanie 8. Załóżmy, że zmienne *x*, *f* i *d* są odpowiednio typów *int*, *float* i *double*. Ich wartości są dowolne, ale *f* i *d* nie mogą równać się $+\infty$, $-\infty$ lub *NaN*. Czy każde z poniższych wyrażeń zostanie obliczone do prawdy? Jeśli nie to podaj wartości zmiennych, dla których wyrażenie zostanie obliczone do fałszu.

1. *x* == (int32_t)(double) *x*
2. *x* == (int32_t)(float) *x*
3. *d* == (double)(float) *d*
4. *f* == (float)(double) *f*
5. *f* == -(-*f*)
6. 1.0 / 2 == 1 / 2.0
7. *d* * *d* >= 0.0
8. (*f* + *d*) - *f* == *d*

¹https://en.wikipedia.org/wiki/Half-precision_floating-point_format