

# Architektury systemów komputerowych 2016

## Lista zadań nr 3

Na zajęcia 14 marca – 17 marca 2016

Jeśli nie stwierdzono inaczej, rozwiązania zadań muszą się trzymać następujących wytycznych:

- Założenia:

- liczby całkowite są w reprezentacji uzupełnień do dwóch,
- wartość logiczna prawdy i fałszu odpowiada kolejno wartościom całkowitoliczbowym 1 i 0,
- przesunięcie w prawo na liczbach ze znakiem jest przesunięciem arytmetycznym,
- dane typu `int` mają  $W$  bitów długości; rozwiązanie musi działać dla dowolnego  $W$  będącego wielokrotnością 8.

- Zabronione:

- wyrażenia warunkowe (`?:`) i wszystkie instrukcje poza przypisaniem,
- operacja mnożenia, dzielenia i reszty z dzielenia,
- operacje logiczne,
- porównania (`<`, `>`, `<=` i `>=`),
- rzutowanie – zarówno jawne jak i niejawne.

- Dozwolone:

- operacje bitowe,
- przesunięcie w lewo i prawo z argumentem w przedziale  $0 \dots W - 1$ ,
- dodawanie i odejmowanie,
- test równości (`==`) i nierówności (`!=`),
- stała  $W$ , stałe własne oraz zdefiniowane w pliku nagłówkowym `<limits.h>`

**Zadanie 1.** Jak szybko podzielić liczbę przez stałą 3? Można używać mnożenia.

**Zadanie 2.** Standard IEEE 754-2008 definiuje liczby zmiennopozycyjne o szerokości 16-bitów. Zapisz ciąg bitów reprezentujący liczbę  $1.5625 \cdot 10^{-1}$ . Porównaj zakres liczbowy i dokładność w stosunku do liczb zmiennopozycyjnych pojedynczej precyzji (`float`).

**Zadanie 3.** Oblicz ręcznie  $3.984375 \cdot 10^{-1} + 3.4375 \cdot 10^{-1} + 1.771 \cdot 10^3$  używając liczb w formacie z poprzedniego zadania. Zapisz wynik binarnie i dziesiętnie. Czy wynik się zmieni jeśli najpierw wykonamy drugie dodawanie?

**Zadanie 4.** 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`

**Zadanie 5.** Reprezentacja liczby zmiennoprzecinkowej typu `float` została załadowana do zmiennej `x` typu `int32_t`. Podaj algorytm mnożenia `x` przez  $2^i$ . Uwzględnij przypadki brzegowe — tj. kiedy zmienna `x` ma wartość  $NaN$ ,  $\pm\infty$ ,  $\pm 0$  lub jest liczbą zdenormalizowaną.

**UWAGA:** Należy podać algorytm, zatem dozwolona jest cała składnia języka C bez ograniczeń z nagłówka listy zadań.

**Zadanie 6.** Reprezentacje binarne liczb zmiennoprzecinkowych `xf` i `yf` typu `float` zostały załadowane odpowiednio do zmiennych `x` i `y` typu `uint32_t`. Podaj wyrażenie, które:

- zmieni znak liczby `x`,
- obliczy wartość bezwzględną z `x`,
- zwróci wartość logiczną operacji `x == y`,
- zwróci wartość logiczną operacji `x <= y`.

Pamiętaj, że dla liczb zmiennopozycyjnych w standardzie IEEE 754 zachodzi  $-0 \equiv +0$ .

**Zadanie 7.** Uzupełnij ciało funkcji zadeklarowanej następująco:

```
/* Skonwertuj reprezentację liczby float do wartości int32_t. */
int32_t float2int(int32_t f);
```

Zaokrąglaj liczbę w kierunku zera. Jeśli konwersja spowoduje nadmiar lub `f` ma wartość  $NaN$ , zwróć `0x80000000`.