

Systemy komputerowe

Lista zadań nr 1

Na zajęcia 28 lutego – 4 marca 2019
(wersja robocza – brak zadania 3)

Zadanie 1. Zadanie składa się z podpunktów:

- (a) Wartość `sizeof(struct A)` wynosi 8 a nie, jak można by się spodziewać 5. Dlaczego?
- (b) Podobnie, `sizeof(struct B)` wynosi 8. Dlaczego? Uwaga: argument jest *bardzo podobny* jak w poprzednim podpunkcie, ale nie dokładnie ten sam.
- (c) Czy kompilator języka C może umieścić zmienną typu `struct A` pod adresem `0x7ffc9e54cb12`?
- (d) Podaj przykład definicji struktury w języku C, która ma różne rozmiary w zależności od kolejności wystąpienia zdefiniowanych pól. Dlaczego kompilator sam nie wybiera optymalnej permutacji pól?

```
struct A {                struct B {
    int8_t a;              int32_t b;
    int32_t b;             int8_t a;
};                          };
```

Wskazówka: CSAPP 3.9, 3.9.3

Zadanie 2. Jakie nieoczekiwane zachowanie przejawiają poniższe fragmenty kodu?

```
unsigned i;                #define DELTA sizeof(int)
for (i = cnt-2; i >= 0; i--) int i;
    a[i] += a[i+1];        for (i = CNT; i-DELTA >= 0; i-= DELTA)
                            . . .
```

Wskazówka: CSAPP 2.2.5

Zadanie 3. Tu znajdzie się treść zadania...

Zadanie 4. Dana jest tablica n elementów typu `int`. Dokładnie jedna wartość występuje w tablicy nieparzyście wiele razy. Podaj algorytm działający w czasie $O(n)$ i dodatkowej pamięci $O(1)$ identyfikujący tę wartość.

Wskazówka: Operacja bitowa.

W zadaniach 5, 6, 7 i 8 wolno używać **wyłącznie** instrukcji przypisania, operatorów bitowych, dodawania, odejmowania i przesunięć bitowych. Wszystkie zmienne mają typ `uint32_t`. Można używać zmiennych tymczasowych oraz stałych.

Zadanie 5. Zmienne i, k spełniają warunek $0 \leq i, k \leq 31$. Napisz fragment kodu, który skopiuje i -ty bit zmiennej x na pozycję k -tą.

Zadanie 6. Napisz fragment kodu, który wyznaczy liczbę zapalonych bitów w zmiennej x .

UWAGA! Oczekiwana złożoność to $O(\log n)$, gdzie n to liczba bitów w słowie. Posłuż się strategią „dziel i zwyciężaj”.

Zadanie 7. Podaj fragment kodu, który oblicza funkcję:

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

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

Zadanie 8. Uzupełnij ciało funkcji zadeklarowanej poniżej.

```
/* Kiedy 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 9. 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.

- $(x > 0) \ || \ (x - 1 < 0)$
- $(x \ \& \ 7) \ != \ 7 \ || \ (x \ \ll \ 29 < 0)$
- $(x * x) \ \geq \ 0$
- $x < 0 \ || \ -x \leq 0$
- $x > 0 \ || \ -x \geq 0$
- $(x \ | \ -x) \ \gg \ 31 == -1$
- $x + y == (\text{uint32_t})y + (\text{uint32_t})x$

Zadanie 10. 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 11. 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?

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