

Systemy komputerowe

Lista zadań nr 1

Na zajęcia 2–5 marca 2020

W zadaniach odnoszących się do języka C wolno używać **wyłącznie** instrukcji przypisania, operatorów bitowych, dodawania i odejmowania, przesunięć bitowych i stałych! Pętle, rozgałęzienia, operatory mnożenia, dzielenia i reszty z dzielenia są **niedozwolone**! Zakładamy, że liczby są typu «uint32_t» – tj. nie posiadają znaku i mają szerokość 32 bitów. Należy wytłumaczyć dlaczego rozwiązanie działa!

W trakcie prezentacji rozwiązań należy zdefiniować i wyjaśnić pojęcia, które zostały oznaczone **wytłuszczoną** czcionką.

Zadanie 1. Napisz fragment kodu (w języku C), który dla zmiennych x i k wykona poniższe operacje:

- wyzeruje k -ty bit zmiennej x ,
- ustawi k -ty bit zmiennej x ,
- zaneguje k -ty bit zmiennej x .

Zadanie 2. Napisz fragment kodu, który dla zmiennych x i y obliczy poniższe wyrażenia:

- $x * 2^y$,
- $\lfloor x/2^y \rfloor$,
- $x \bmod 2^y$,
- $\lceil x/2^y \rceil$.

Zadanie 3. Napisz fragment kodu, który bez użycia dodatkowych zmiennych, zamieni miejscami zawartość zmiennych x i y .

Wskazówka: Spróbuj rozwiązać zadanie samodzielnie, a następnie przeczytaj §2.19 książki „Uczta programistów”.

Zadanie 4. Dotyczy CSAPP3e: Bits, Bytes and Integers, Part 1, slajdy nr 37 i 40.

- Uzasadnij poprawność algorytmu konwersji w -bitowej liczby ze znakiem do liczby $w + k$ bitowej.
- Dla jakich wartości całkowitych reprezentowalnych na $w + k$ bitach obcięcie do w bitów nie powoduje zmiany tych wartości?

Zadanie 5. Napisz fragment kodu, który stwierdza czy dana liczba x nie jest potęgą dwójki.

Zadanie 6. 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 7. 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 8. Napisz fragment kodu, który skonwertuje zmienną x z formatu **little-endian** do formatu **big-endian**. Należy użyć jak najmniejszej liczby operacji bitowych.

Zadanie 9. Jakie nieoczekiwane zachowanie przejawiają poniższe fragmenty kodu? Jak je naprawić?

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

Wskazówka: CSAPP 2.2.5

Zadanie 10. Jaką rolę pełnią **kody sterujące** standardu **ASCII** o numerach 0, 4, 7, 10 i 12?

Zadanie 11. Jakie ograniczenia standardu ASCII przyczyniły się do powstania **UTF-8**? Wyjaśnij zasadę kodowania znaków do postaci binarnej UTF-8 i zapisz poniższy ciąg znaków w systemie szesnastkowym:

Proszę zapłacić 5€!