Wprowadzenie do programowania w języku C
grupa RKr, wtorek 16:15-18:00
lista nr 6 *(na pracownię 19.11.2019) (wersja 2)*

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

[2p | **A]** Przepisz poniższą definicję typu strukturalnego do swojego programu i użyj operatora sizeof(.) aby sprawdzić jaki rozmiar ma typ **Box**. Rozmiary pól składowych sumują się do 8 lub 12 bajtów, jednak cała struktura może być większa. Zmień uporządkowanie pól w strukturze, aby osiągnąć oczekiwany rozmiar.
*wskazówka: Zapoznaj się z pojęciami **natural alignment** i **padding**, jeśli chcesz wiedzieć skąd to zjawisko.*

```
tyepdef unsigned char byte;
struct _Box_ { //           Windows // Linux    (these are typical values for Windows/Linux x64,
    byte b0;   // sizeof =        1 // 1           they may look different on your system/machine)
    long  l;   // sizeof =        4 // 8
    byte b1;   // sizeof =        1 // 1
    short s;   // sizeof =        2 // 2
} Box;         //  total =        8 // 12    ...is it?
```

[3p | **B]** Napisz funkcję „**void printCPU_Bitness**()", która w zależności od architektury procesora, który wykonuje twój program, wypisze komunikat „[CPU]Bitness: 64-bit" lub „[CPU]Bitness: 32-bit".
*wskazówka: Możesz wykorzystać operator **sizeof**(.), zastanów się do jakiego typu go zaaplikować.*

[5p | **C]** Napisz funkcję „**void printCPU_Endianness**()", która zgodnie z architekturą procesora wykonującego twój program wypisze komunikat „[CPU]Endiannes: little-endian" lub „[CPU]Endiannes: big-endian".
*wskazówka: Potraktuj **long** jako tablicę 4|8-elementową T i sprawdź gdzie w pamięci znajduje się najsłabszy bajt.*
*Możesz użyć rzutowania „T = (**byte***)((**long***)&n)" dla liczby zdefiniowanej jako „**long** n = 1;".*
*Zamiast rzutowania postaraj się jednak użyć unii bitowej, słowo kluczowe **union**.*
*Zdefiniuj typ **byte** jako **unsigned char** poprzez **typedef.***

Zadanie 2. [10p]
Zaprogramuj strukturę danych zbudowaną w oparciu o wskaźniki, która przechowuje pary <key, val> w sposób uporządkowany względem klucza (key), a także pozwala na operacje wstawiania, usuwania oraz wyszukiwania elementów, wszystkie operacje powinny działać względem klucza. Przyjmij, że klucz (key) jest typu **int** (zdefiniuj aliasujący typ **Tkey**), a wartość (val) jest typu **char** albo **float** (zdefinuj aliasujący typ **Tval**).

```
void insert(?, Tkey key, Tval val);
void delete(?, Tkey key);
Tval search(?, Tkey key);
```

Wybierz jedną z poniższych struktur:
[**A**] lista pojedynczo wiązana (*ang. singly-linked list*), inaczej lista jednokierunkowa
[**B**] lista podwójnie wiązana (*ang. doubly-linked list*), inaczej lista dwukierunkowa
[**C**] binarne drzewo poszukiwań (*ang. binary search tree*)

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