

Оформяне 2

Програмиране за 10 клас

28 юни 2017г. (вариант А)

Време: 80 мин

Важно: Задачите, които предавате, трябва да се компилират! Некомпилиращи се файлове получават 0 точки.

Задача 1

Да се дефинира структурата **book_t**, която съдържа в себе си име (до 50 символа), година на издаване и брой продадени копия. Да се дефинира функцията **float estimate_rating(struct book_t)**, която казва ориентировъчен рейтинг на книгата, като го пресмята по формулата (годината * копията) / дължината на името.

Да се дефинира функцията **int rating_between(struct book_t[10], int min, int max)**, която намира и връща броя книги с рейтинг между **min** и **max**.

Задача 2

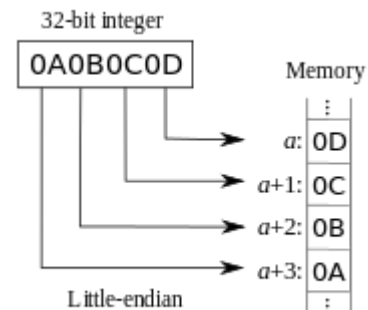
Дефинирайте структурата **book_t** от горната задача. Да се дефинира структурата **bookstore_t**, която съдържа масив до 10 книги и текущия брой книги в книжарницата.

Да се дефинира функцията **int sorted_add(struct bookstore_t*, struct book_t)**, която добавя книгата към книжарницата, ако тя не е пълна. Приемаме, че книгите са сортирани по години във възходящ ред и нашата книга трябва да се добави на първата възможна позиция, така че да не нарушава реда на годините. Функцията да връща индекса на който е добавена книгата, а ако няма място -1.

Задача 3

Направете четене на символи до достигане на EOF, които представляват байтове от паметта (от 0 до F), записани в little-endian формат т.е. числото 'ABCD0123' (изходния низ) е записано като 'DCBA3210' (входния низ) в паметта, както е показано на диаграмата долу. Приемете, че броят символи винаги е кратен на 4.

Дефинирайте функцията **char* fix_endian(char*)**, която "обръща" всяка 4ка символи и връща първоначалния низ, преди той да е бил записан в паметта.



Задача 4

Вектор е структура от данни, която представлява самоообразмеряващ се масив. Да се дефинира вектор (vector), който поддържа функциите за добавяне - push и за премахване - pop на елемент, в края на масива.

Дефинирайте функциите **create_vector** и **destroy_vector**, които съответно създават празен вектор и изтриват даден вектор.

Дефинирайте функциите **push_front** и **pop_front**, които добавят или махат елемент от вектора от едната му страна.

Дефинирайте функцията **at**, която по даден индекс връща елемента на дадения индекс. Ако индексът е невалиден да се върне -1.

Пояснение:

Ако имаме празен вектор и добавим (push) 1, след това - 2, след това - 6 и след това извикаме pop трябва да получим числото 6, тъй като това е последният добавен елемент и във вектора да имаме само 1 и 2.