

Minikurs języka C

Lista zadań nr 1

Na zajęcia 4 marca 2020
grupa PWit

Za każde zadanie otrzymać można max. 10pktów. Do każdego rozwiązania dostarcz *testy jednostkowe* wykonane za pomocą *asercji* (makro `assert` z pliku `assert.h`).

Zadanie 1. Zaimplementuj następujące funkcje:

1. `int max(int array[], size_t arr_size)` — zwracającą maksymalną wartość występującą w `arr_size`-elementowej tablicy `array`.

Uwaga: Jest przynajmniej jeden przypadek brzegowy, który trzeba umieć obsłużyć.

2. `int max2(int array[][77], size_t side_size)` — zwracającą maksymalną wartość występującą w 2-wymiarowej prostokątnej tablicy `array` o wymiarach `arr_size` na 77.
3. Zaimplementuj “generyczną” wersję funkcji `max` z pierwszego podpunktu. Funkcja ma teraz sygnaturę `my_type max(my_type array[], size_t arr_size)`, gdzie typ danych `my_type` zdefiniowany jest dyrektywą `typedef`. Załóż, że programista korzystający z Twojej funkcji `max` dostarcza funkcję `my_type geq(my_type a, my_type b)`, zwracającą większy z elementów `a` i `b`.

Zadanie 2 (Autor: MGa). Kolejka FIFO (First In First Out) to struktura danych która ma dwie operacje: wzięcie elementu (`pop`) i na włożenie elementu do kolejki (`push`), ponadto jeśli dwa elementy zostały włożone do kolejki, to operacja `pop` wcześniej zwróci ten który został wcześniej włożony. W zadaniu należy napisać kolejkę która do której można będzie wkładać następujące elementy:

- Napisy(tablice typu `char`)
- Liczby całkowite
- Pary liczb całkowitych
- Liczby zmiennoprzecinkowe

Zaprojektuj strukturę która będzie mogła przechowywać każdy z tych typów (dobrym pomysłem będzie trzymanie zmiennej mówiącej o przechowywanym typie i pole składowe dla każdego z typów). Kolejkę powinienes zrealizować za pomocą tablicy struktur, dodatkowo funkcja `push` powinna pytać się użytkownika o typ i wartość dodawanych danych, a funkcja `pop` powinna wypisywać usuwany element. Zaprezentuj działanie kolejki, tzn. napisz program który w nieskończonej pętli pyta się użytkownika czy chce dodać czy usunąć element i w zależności od decyzji wywołuje jedną z funkcji.

Uwaga: Twoja implementacja powinna wykrywać próbę wykonania operacji `pop` na pustym stosie oraz `push` na pełnym.

Zadanie 3. Zadanie domowe. Pojawi się w systemie SKOS.