

Wprowadzenie do programowania w języku C

grupa RK, wtorek 16:15-18:00

lista nr 2 (na pracownię 22.10.2019), wersja nr 2

Zadanie 1. [10p (lab) | 5p (dom)] *wskazówka: kolejność sub-zadań dowolna, ale warto zacząć od punktu [A] !!!*

[2p | A] Dla każdego elementu X zadania {B, C, D, E}, zdefiniuj na początku pliku makro TASK_CASE_X i opakuj w makro warunkowe. W funkcji main(), i poza, skorzystaj z tych makr, aby [de]aktywować na etapie kompilacji elementy będące w jednym pliku „.c”. Nie wolno deaktywować kodu komentarzami /* */ lub //.

wskazówka: Użyj dyrektyw #if 0 / #if 1 / #endif oraz #ifdef / #ifndef lub #if defined(?) / #if !defined(?).

[2p | B] Napisz w jednym pliku dwa makra o innych definicjach, ale o tej samej nazwie, uniknij kolizji nazw.

wskazówka: Postaraj się anulować definicję makra dyrektywą preprocesora #undef.

[2p | C] Napisz makro NUM_ELEM(t), obliczające ilość elementów w tablicy alokowanej statycznie,

np. „int T[] = {6, 9};”, „char T[] = {'L', 'U', 'N', 'O', 'K', 'H', 'O', 'D'};”

wskazówka: Skorzystaj z operatora sizeof(?), makro powinno zadziałać niezależnie od typu elementów tablicy.

[2p | D] Napisz makro DECL_INIT(name, type, val), które zawołane tak „DECL_INIT(x, int, 7);”,

wstrzyknie do kodu deklarację „int x_int = 7;”, analogicznie gdy zamiast int będzie float, lub inny typ.

wskazówka: Może się przydać operator ## do konkatencji tokenów.

[2p | E] Napisz funkcję revlog2(?), która potraktuje swój argument jako logarytm pewnej liczby i w wyniku da Ci tę liczbę. Nie można używać funkcji bibliotecznych.

dla ambitnych: Implementacja w czasie stałym, O(1), wystarczy dokładnie jedna instrukcja.

Zadanie 2. [10p] *wskazówka: Uważaj na reguły nierozstrzygalne jedną literą, np. [n » H » n] vs [nie » He » niye].*

Zignoruj zachowywanie czarnych/niebieskich wielkich liter jeśli to trudne lub niemożliwe.

Zaprogramuj transliterator, który słowo w alfabecie **polskim** transliteruje na uproszczony alfabet **rosyjski** (litery łacińskie i cyfry arabskie), a następnie na głoski/fonemy **angielskie**. Przyjmij reguły: [cz » 4 » ch]

[sz » w » sh] [b » 6 » b] [h » X » kh] [ż » } { » zh] [i » N » ee] [nie » He » niye] [e » e » ye] [w » B » v]

[r » p » r] [u » y » u] [n » H » n] [s » C » s] [k » K » k] [bie » 6e » biye] [rie » pe » riye] [L » JT » L] [t » T » t],

litera bez reguły jest niezmienna względem alfabetu. Staraj się utrzymywać rozmiar liter czerwonych tak jak podano w regułach, wyjątkiem mogą być pierwsze litery w nazwach własnych, imionach, etc.

Program musi wypisać obydwie transliteracje, tj. rosyjską i angielską, przy czym rosyjska musi być utworzona ze słowa polskiego, a angielska ze słowa rosyjskiego. Przykładowe transliteracje do weryfikacji programu:

Bielka i Strielka » 6eJTKa N CTpeJTKa (rus. Белка и Стрелка) » Biyelka ee Striyelka (suczki w sat. Sputnik 5)

Wera Breżniewa » Bepa 6peJ {HeBa (rus. Вера Брежнева) » Vyera Bryezhniyeva (młodsza pani)

czebureki » 4e6ypeKN (rus. чебуреки) » chyeburyekee (duże pierogi)

babuszka » 6a6ywkA (rus. бабушка) » babushka (starsza pani)

uszanka » ywaHKA (rus. ушанка) » ushanka (zimowa czapka)

haski » XaCKN (rus. хаски) » khaskee (zimowy pies)

Zadanie 3. [10p] Dostępne w serwisie SKOS, sprawdzane przez automat.