

Minikurs języka C

Lista zadań nr 2

Na zajęcia 1 kwietnia 2020
grupa AKu

Za każde zadanie można otrzymać od 0 do 10 punktów.

Zadanie 1. Przez $F(n, k)$ oznaczmy ilość n -cyfrowych liczb naturalnych o sumie cyfr równej k .

1. (3 pkt) Zaimplementuj funkcję `int F(int n, int k)` rekurencyjnie obliczającą wartość $F(n, k)$ ¹.
2. (4 pkt) Zauważ, że funkcja z podpunktu 1. może obliczać te same wartości wielokrotnie, co może prowadzić do wykładniczego czasu wykonania programu. Ponadto obliczone wyniki szybko wyczerpują zakres typu `int`.

Zaimplementuj rekurencyjną funkcję `int FMem(int n, int k)`, która oblicza wartość $f(n, k) \bmod 10^9 + 7$ przy pomocy memoryzacji. W tym celu zdefiniuj pomocniczą tablicę, w której będziesz przechowywać wyniki wywołań rekurencyjnych. Sprawdź czy twoja implementacja funkcji `FMem` faktycznie działa szybciej od funkcji `F` z podpunktu 1.

3. (3 pkt) Zaimplementuj funkcję `void FGen(int n, int k)`, która wypisuje na standardowe wyjście wszystkie n -cyfrowe liczby o sumie cyfr równej k .

Przetestuj napisane funkcje przy pomocy asercji.

Zadanie 2. [Autor zadania: TDr] Przyglądnij się dołączonemu do listy plikowi `decimalio.c` i zrozum, jak działają obecne w nim funkcje. Utwórz dla niego odpowiedni plik nagłówkowy `decimalio.h`, który nie będzie załączał `stdio.h`. Utwórz plik `main.c`, który również nie będzie załączał `stdio.h`, wczyta liczbę n i wypisze w osobnych liniach n pierwszych kwadratów liczb naturalnych (dla 4 wypisze liczby 0, 1, 4 i 9). W pliku `compile.sh` zapisz polecenie, którym kompilowany jest cały program. Poleceniem `chmod +x compile.sh` można na Linuksie nadać mu prawo do wykonywania i uruchamiać `./compile.sh`.

Zadanie 3. Pojawi się w systemie SKOS.

¹Aby otrzymać rekurencyjny wzór na $F(n, k)$ ustal jedną z cyfr liczby o długości n na wszystkie możliwe sposoby.