



Четврто вежбање

Вежба 1

Илустрација рада рекурзивних функција. Направите нови пројекат и укључите следеће датотеке: „functions.c“, „functionEg.h“ и „functionEg.c“. Избилдујте и покрените програм.

У првом делу вежбе неопходно је имплементирати рекурзивну функцију која рачуна факторијел датог броја. Декларација функције је дата у датотеци „functionEg.h“. Формула за рачунање факторијела је дата у наставку.

$$\begin{aligned} n! &= (n-1)! * n && \text{if } n > 0 \\ &= 1 && \text{if } n = 0 \end{aligned}$$

Други део вежбе се односи на имплементацију рекурзивне функције која враћа Фибоначијев број за било који задати број из Фибоначијевог низа. Декларација функције је дата у „functionEg.h“ датотеци.

Фибоначијев низ бројева је низ бројева са особином да сваки члан низа, осим прва два, једнак је збиру претходна два члана:

$$F_n = F_{n-1} + F_{n-2}, \quad F_0 = 0 \text{ and } F_1 = 1$$

Након имплементација рекурзивних функција неопходно је имплементирати наведене две функције без коришћења рекурзије.

Покренути извршавање свих функција са истим улазним параметром. Упоредити време извршавања рекурзивне и функције имплементиране без рекурзије.

Вежба 2

У датотеци program.c налази се програм сличан ономе који смо на претходних неколико вежби користили. Проширити функцију print да одређује који су парни



бројеви у низу мањи од вредности дефинисане константом `treashold`. Направити да те бројеве и њихове индексе некако враћа функцији `main`, а да их функција `main` испише на екран. Затим, полазећи од тако измењеног програма направити две његове различите варијанте: једну где функције немају никакве параметре и повратне вредности, а другу где се глобалне променљиве уопште не користе (дакле све се преноси кроз параметере или повратне вредности).

Вежба 3

Вежба илуструје комбиновање С и асемблер функција у оквиру једног пројекта. У овој вежби је показано како се врши прослеђивање аргумената функцијама и коришћење `ASM` директива.

1. Направите нови С пројекат и додајте `e_functions.c`, `defines.h` и `functionExamples.S`
2. Подесите одговарајућу “include” путању неопходну за асемблирање `functionExamples.S` (Project properties -> C/C++ Build -> Settings -> GCC Assembler -> General -> Include Paths)
3. Избилдујте и покрените програм

Анализирајте код