

Домаћи

Пре него што почнете прочитајте поглавље [УПУТСТВО](#) на крају овог документа. Ту ћете пронаћи упутства како да попуните овај документ и како треба да доставите своја решења задатака из овог домаћег.

ЗАДАЦИ

- I. Прочитајте поглавља 2 и 3 књиге Бјарнеа Строуструпа „*Programming Principles and Practices C++*“. Научите значења свих појмова из тих поглавља одговорите на следећа питања:

а) Набројте четири дела дефиниције функције:

- 1.
 - 2.
 - 3.
 - 4.

б) Приликом учитавања са стандардног улаза, шта означава крај једног уноса за целе бројеве и знаковне низове?

с) Шта је објекат?

д) Шта је литерал? Наведите неколико типова литерала.

- е) Шта су променљиве? Наведите основне типове променљивих и њихове уобичајене величине.

- ф) Која је разлика између операција = и ==? Ког типа је њихов резултат?

- г) Шта је дефиниција, а шта декларација?

- х) Шта је иницијализација и по чему се разликује од доделе?

- и) Како се знаковни низови надовезују у Це++ језику?

- ј) Шта је типска безбедност и зашто је важна?

- к) Да ли је смештање вредности **double** типа у променљиву **int** типа типски безбедно? Зашто?

II. Напишите добро структуриран и коментарисан C++ програм који садржи три функције описане у наставку. Када се овде каже „хексадецимални број“ мисли се на стринг са хексадецималним записом броја.

- a) Функција 1 – прима неозначени цео број и враћа стринг са записом тог броја у хексадецималној форми. Функција треба да баци изузетак у случају неодговарајућег улаза.
- b) Функција 2 – прима два хексадецимална броја (стрингове са хексадецималним записом броја) и враћа њихов збор исто у хексадецималном облику. Функција треба да баци изузетак у случају неодговарајућег улаза.
- c) Функција 3 – прима хексадецимални број и враћа целобројну вредност која одговара том хексадецималном запису. Функција треба да баци изузетак у случају неодговарајућег улаза.
- d) Позовите ове функције из мејн функције. Употребите трај (try) и кеч (catch) механизме.
- e) Проверите исправност вашег програма помоћу примера:
 - a. исправног улаза
 - b. неисправног улаза
 - c. празног улаза

- III. Напишите програм који очекује од корисника да унесе две вредности истог основног типа (**short**, **int** или **double**). Те вредности треба сместити у променљиве **val1** и **val2**. Затим треба одредити мањи и већи од та два броја, њихов збир, разлику, производ и количник, и исписати те резултате на стандардни излаз. Употребите три функције, по једну за сваки тип променљивих. Омогућити кориснику да изабере коју од те три функције (та три типа) жели.

- IV. Напиши програм који рачуна Фибоначијеве бројеве. Формула за генерисање Фибоначијевих бројева дата је у наставку:

$$F_n = F_{n-1} + F_{n-2},$$

где је $F_1 = 1$ и $F_2 = 2$.

На пример, $F_3 = F_2 + F_1 = 2 + 1 = 3$.

Приметите да ће у једном тренутку Фибоначијеви бројеви бити превелики за **int** (или било који други) тип и доћи ће до прекорачења. Размислите како да прекинете рачунање бројева када дође до прекорачења, или непосредно пре него што до прекорачења дође, и то имплементирајте.

Напишите три посебне функције за три различита типа: **int**, **long** и **long long**.

Питања:

- a. Колико првих Фибоначијевих бројева можемо сместити у тип **int**?

Која је вредност највећег?

- b. Колико првих Фибоначијевих бројева можемо сместити у тип **long**? Која је вредност највећег?

- c. Колико првих Фибоначијевих бројева можемо сместити у тип **long long**? Која је вредност највећег?

УПУТСТВО:

Напишите одговоре у одговарајућа поља. Проширите величину поља ако вам треба више места. Решења за задатке II, III и IV, треба да буде приложено уз овај документ. Све то спакујте у Зип архиву. Архиву треба назвати овако:

ИМЕ_ПРЕЗИМЕ_ИНДЕКС_domaci.zip

(Где *ИМЕ*, *ПРЕЗИМЕ* и *ИНДЕКС* треба заменити са вашим именом, презименом и бројем индекс)

Архива треба да садржи:

Датотеке:

domaci.docx (са одговорима на питања)

Директоријуме:

II – који садржи пројекат и изворни код за задатак 2

III – који садржи пројекат и изворни код за задатак 3

IV – који садржи пројекат и изворни код за задатак 4

У решењима задатака треба користити само оне елементе језика Це++ које смо спомињали током прве три недеље на предавањима. Циљ није да се размећете напредним знањем Це++ синтаксе.

ВАЖНО

Пројектни директоријуми (*II*, *III* и *IV*) **НЕ СМЕЈУ** садржати извршне или објектне датотеке!!! Дозвољене су искључиво следеће датотеке:

.vcxproj
.sln
.filters
.cpp
.h
.hpp

Изооставити аутоматски направљене директоријуме: *Debug*, *Release*, *ipch* и сл.