За потребе ове вежбе:

- У развојном окружењу CS50 IDE направити нови директоријум *lab03*.
- Преузети кодове и прекопирати их у директоријум *lab03* у вашем развојном окружењу.
- Пре превођења и покретања било ког кода потребно је позиционирати се у одговарајући директоријум у терминалу.
- За превођење користити команду: ***g++ [naziv_datoteke] -o[naziv_programa]***
- У случају да је потребно покренути програм уз помоћ алата за дебаговање (*debug50)* неопходно је у приликом превођења додати и опцију: *-ggdb3*
- Као коначно решење ове вежбе поставите датотеке *MyBigInt.h*, *BigData.h*, *zadatak2.cpp* и *MyUniquePtr.h*

Задатак 1

Почните од класе MyBigInt од претходног дана и прерадите је тако да динамички заузима меморију за цифре, слично томе како то раде std::vector и std::string. Након тога додајте мув конструктор и мув доделу.

Задатак 2

Циљ овог задатка је имплементирати мув конструктор и мув оператор доделе за тип BigData (у BigData.h, у директоријуму zadatak2). Након инмплементације мув функционалности потребно је анализирати примере из main.cpp датотеке. Такође, потребно је анализирати и време извршења користећи std::chrono.

Задатак 3

Користећи функцију **type_name** дату у type_name.hpp, проверите шта ће бити тип T у случајевима датим у следећој табели (тј. попунити табелу).

template< class T>	5	int a;	int& r=a;	int&& rr=5;	const int ca;	const int& cr=ca;
foo(T x)	T-?					
foo(T& x)						

foo(const T& x)						
foo(T&& x)						

Такође, попунити исту табелу али овога пута са податком ког типа ће бити променљива **x** ако је декларисана са кључном речи **auto**.

auto	5	int a;	int& r=a;	int&& rr=5;	const int ca;	const int& cr=ca;
auto x	тип x - ?					
auto& x						
const auto& x						
auto&& x						

Ево примера како се **type_name** користи:

std::cout << type_name<T>() << std::endl;

где је **T** тип чије име желимо да испишемо, или овако:

std::cout << type_name<decltype(x)>() << std::endl;

где је **x** променљива чији тип желимо да испишемо.

Код овог задатка немате резултат који треба да се качи на Канвас.

Задатак 4

Циљ овог задатка је имплементирати свој тип јединственог показивача (MyUniquePtr) који ће имати основне могућности библиотечког типа unique_ptr. У main.cpp датотеци налази се код који користи unique_ptr. Ваш тип јединственог показивача би требало да може заменити unique_ptr у том коду.

У датотеци MyUniquePtr.h дефинисати шаблонски тип MyUniquePtr. Претпроцесорским директивама #define на почетку датотеке main.cpp заменити

unique_ptr са MyUniquePtr. Дати код треба да се након тога преводи и понаша исто, а закоментарисани код (блок „GRESKE“) треба да проузрокује грешку током превођења. Обратити пажњу и на @TODO коментар на крају функције main.

Напомена: Оператор -> је мали изузетак. Пише се као метода и не прима параметре. Треба да врати показивач на тип који има поље које одговара називу са десне стране стрелице.