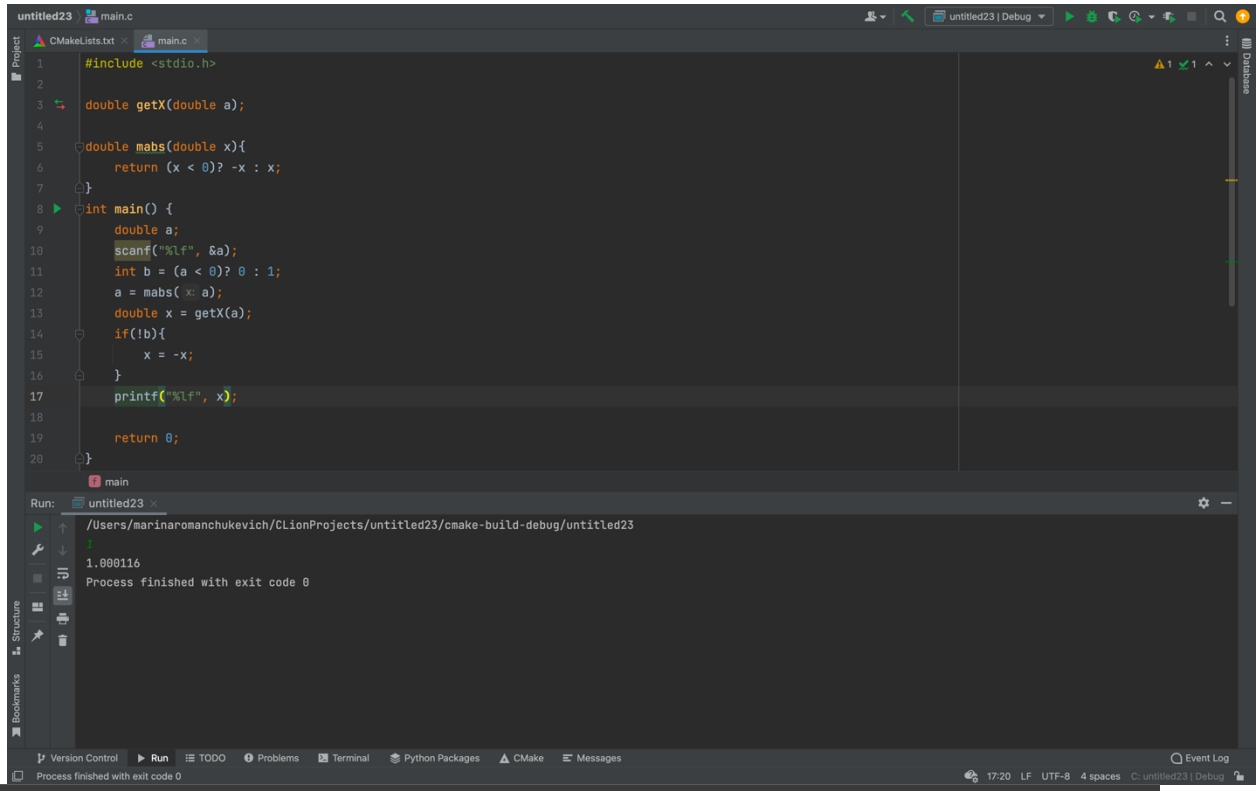
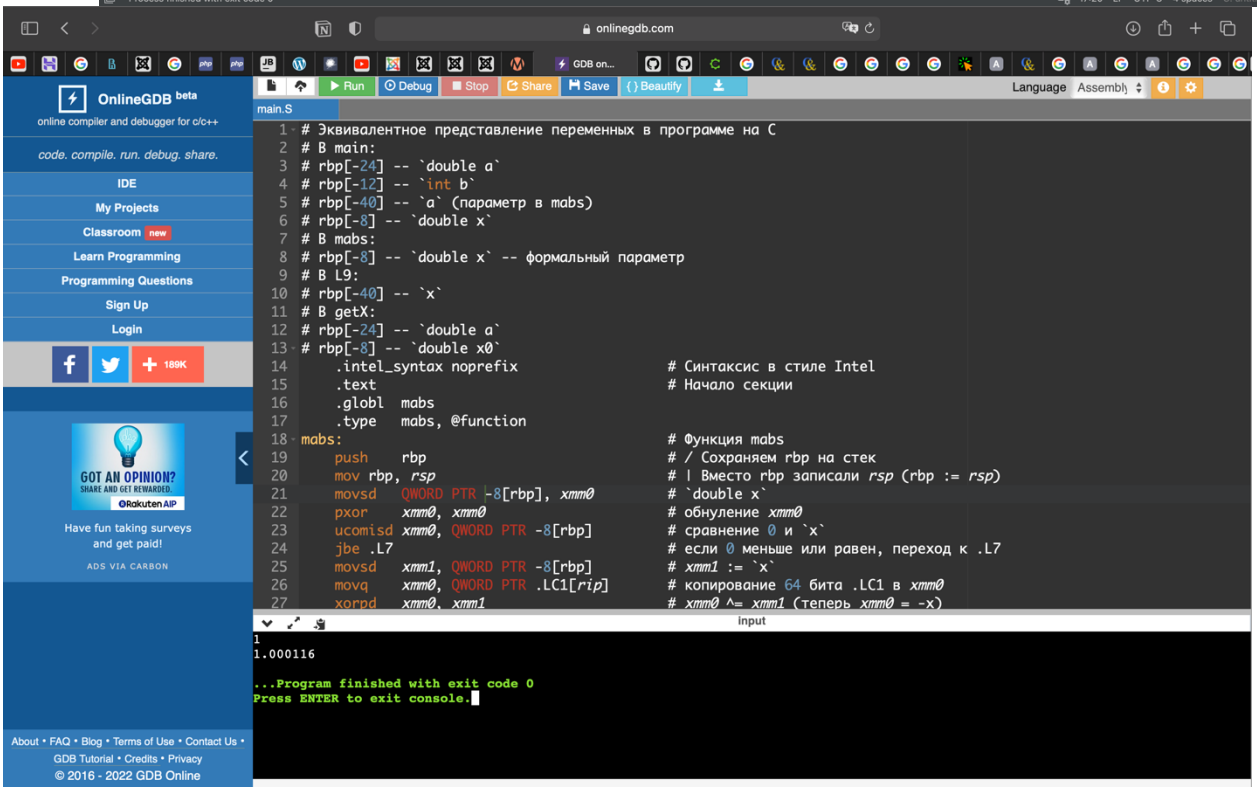
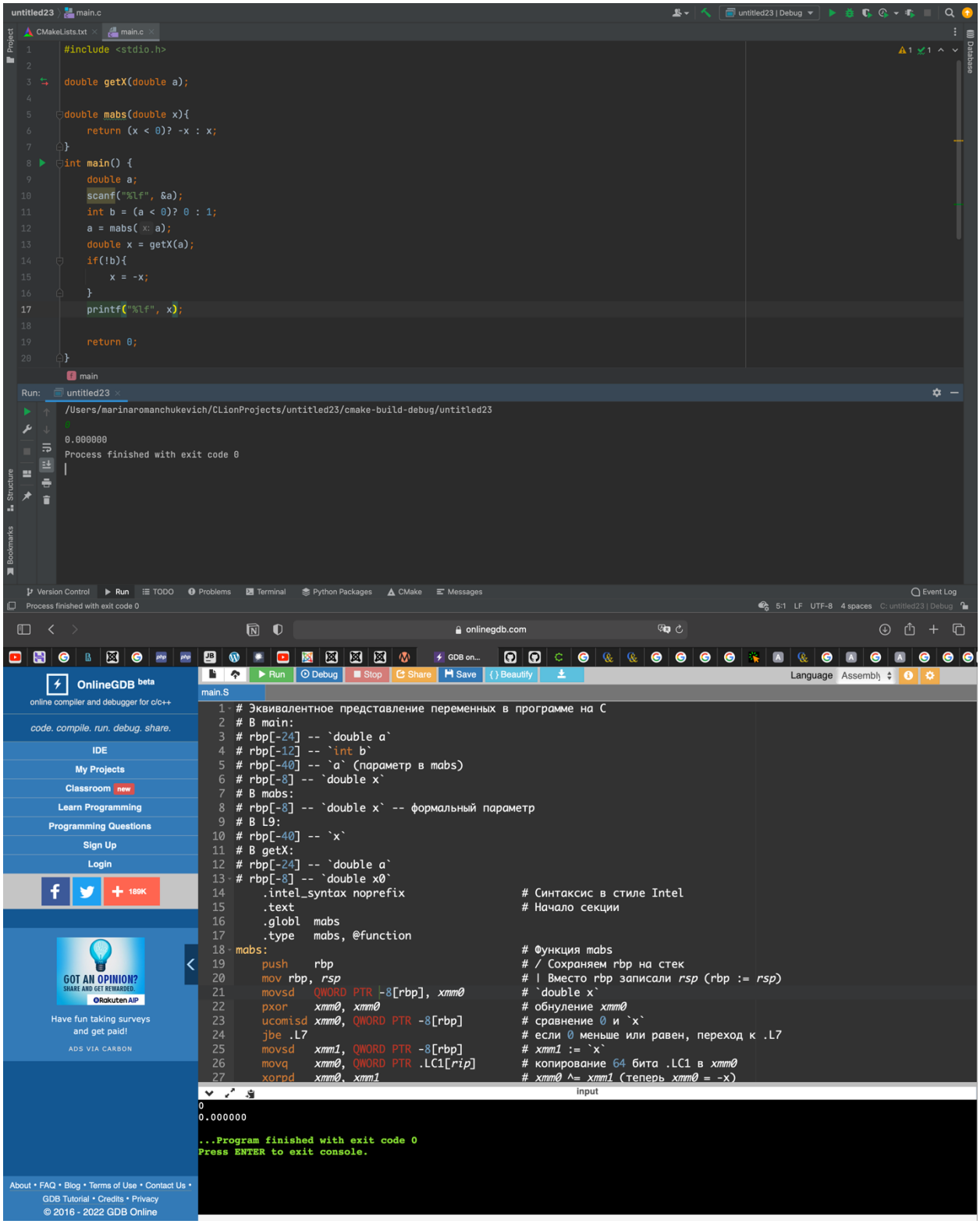
1. Романчукевич Марина Александровна
2. БПИ216
3. Вариант 26 Разработать программу вычисления корня пятой степени согласно быстро сходящемуся итерационному алгоритму определения корня n -той степени с точностью не хуже 0,1%.
4. Тест 1: 1

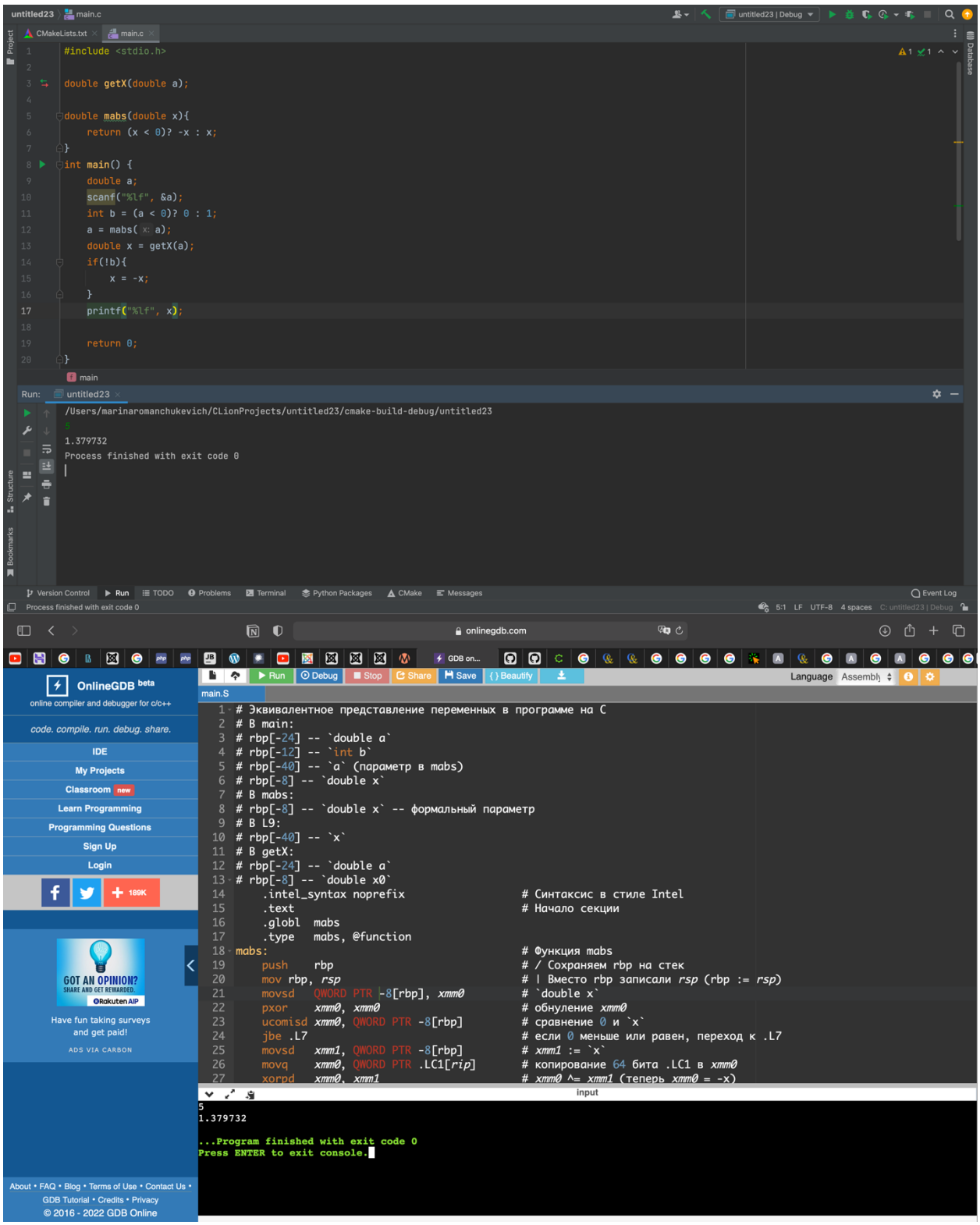




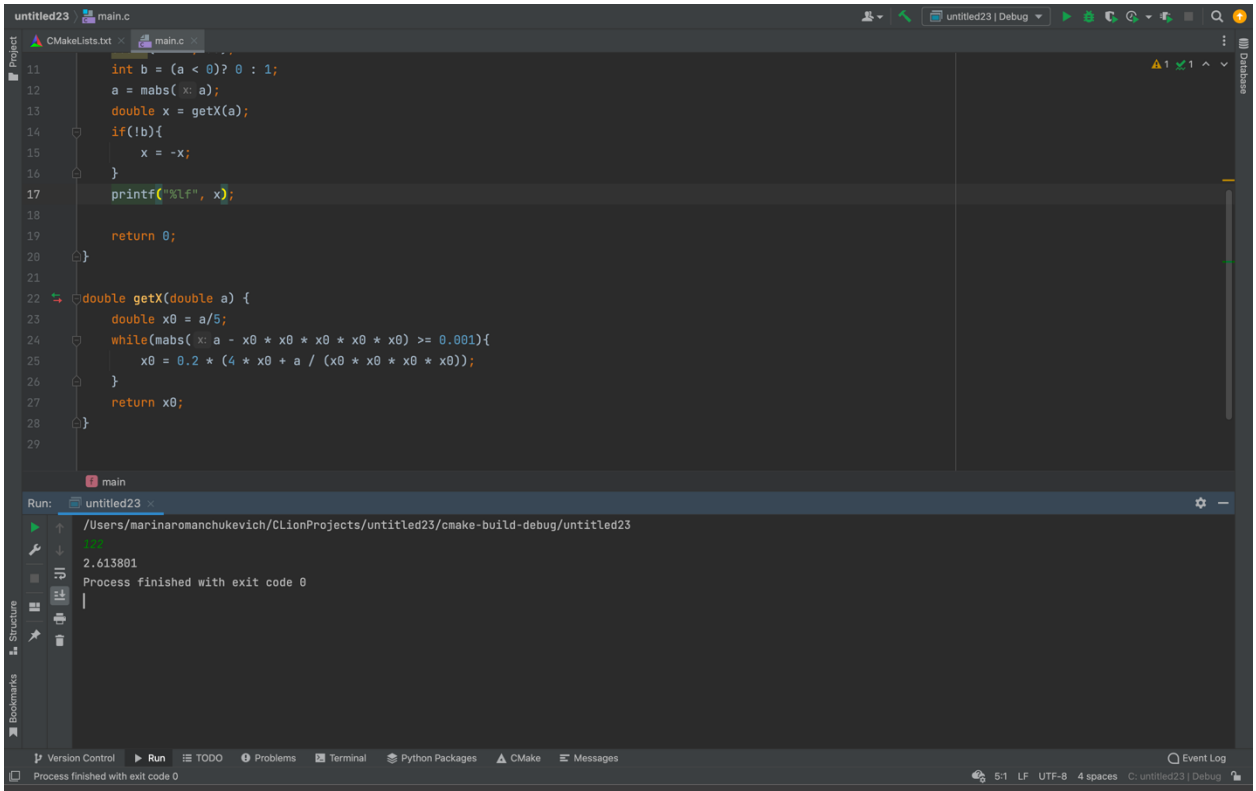
Тест 2: 0

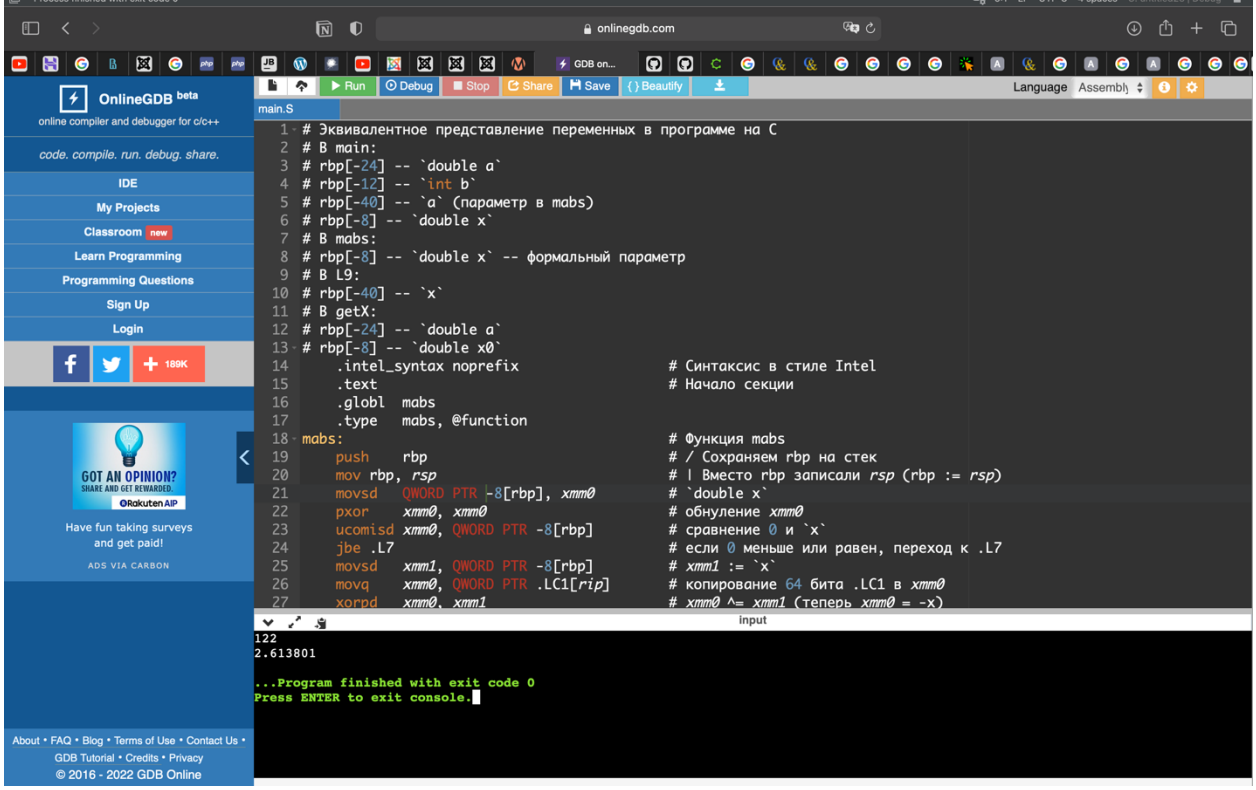


Тест 3: 5

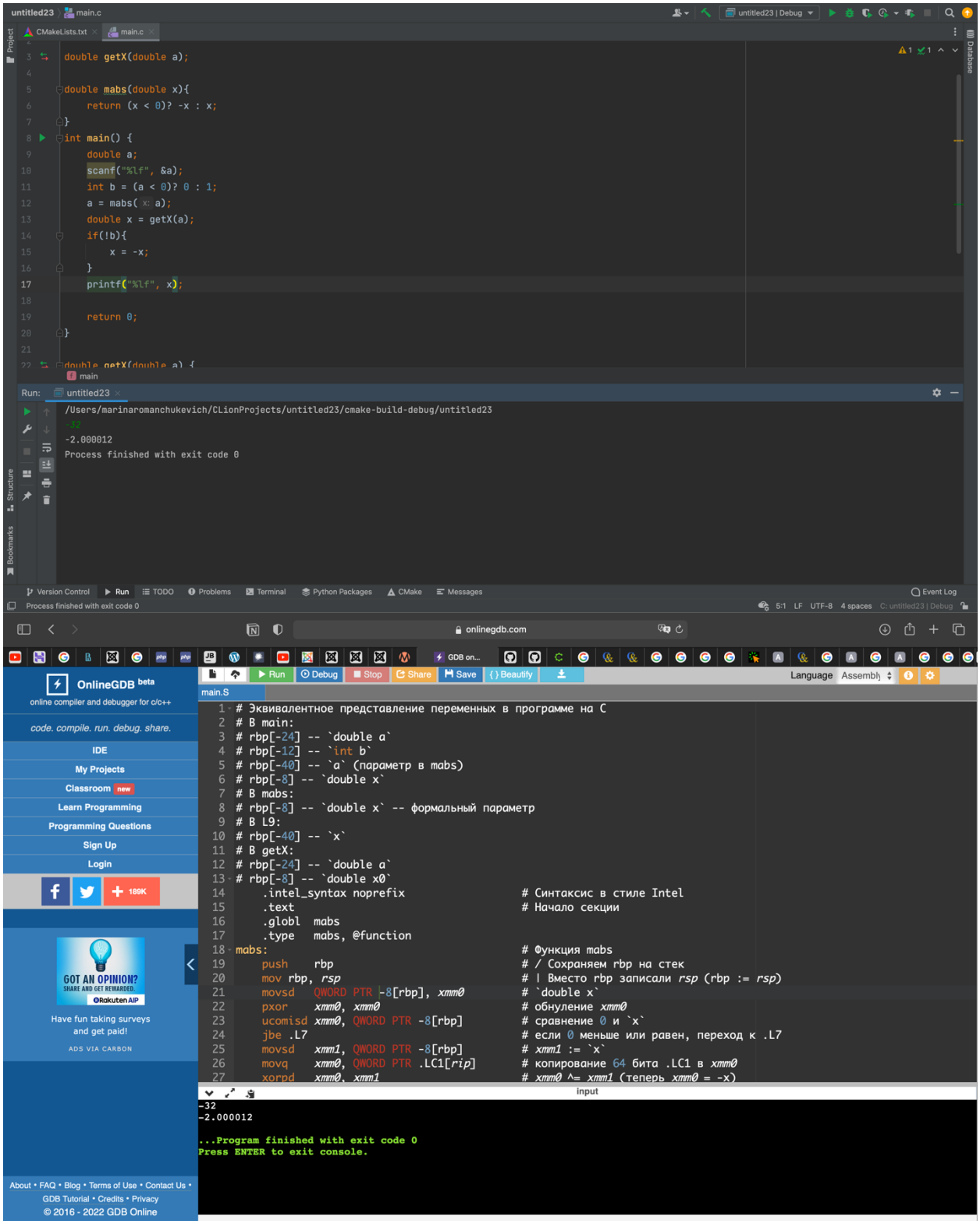

untitled23
main.c
CMakeLists.txt
#include <stdio.h>
double getX(double a);
double mabs(double x){
return (x < 0)? -x : x;
}
int main() {
double a;
scanf("%lf", &a);
int b = (a < 0)? 0 : 1;
a = mabs(x: a);
double x = getX(a);
if(!b){
x = -x;
}
printf("%lf", x);
return 0;
}
Run:
untitled23
/Users/marinaromanchukevich/CLionProjects/untitled23/cmake-build-debug/untitled23
5
1.379732
Process finished with exit code 0
Version Control
Run
TODO
Problems
Terminal
Python Packages
CMake
Messages
Event Log
onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
189K
GOT AN OPINION?
SHARE AND GET REWARDED
Rakuten AIP
Have fun taking surveys
and get paid!
ADS VIA CARBON
About • FAQ • Blog • Terms of Use • Contact Us •
GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online
Run
Debug
Stop
Share
Save
Beautify
Language
Assembly
main.S
Эквивалентное представление переменных в программе на C
В main:
rbp[-24] -- `double a`
rbp[-12] -- `int b`
rbp[-40] -- `a` (параметр в mabs)
rbp[-8] -- `double x`
В mabs:
rbp[-8] -- `double x` -- формальный параметр
В L9:
rbp[-40] -- `x`
В getX:
rbp[-24] -- `double a`
rbp[-8] -- `double x0`
.intel_syntax noprefix # Синтаксис в стиле Intel
.text # Начало секции
.globl mabs
.type mabs, @function
mabs: # Функция mabs
push rbp # / Сохраняем rbp на стек
mov rbp, rsp # | Вместо rbp записали rsp (rbp := rsp)
movsd QWORD PTR -8[rbp], xmm0 # `double x`
pxor xmm0, xmm0 # обнуление xmm0
ucomisd xmm0, QWORD PTR -8[rbp] # сравнение 0 и `x`
jbe .L7 # если 0 меньше или равен, переход к .L7
movsd xmm1, QWORD PTR -8[rbp] # xmm1 := `x`
movq xmm0, QWORD PTR .LC1[rip] # копирование 64 бита .LC1 в xmm0
xorpd xmm0, xmm1 # xmm0 ^= xmm1 (теперь xmm0 = -x)
input
5
1.379732
...Program finished with exit code 0
Press ENTER to exit console.

Тест 4: 122


int b = (a < 0)? 0 : 1;
a = mabs(x: a);
double x = getX(a);
if(!b){
x = -x;
}
printf("%lf", x);
return 0;
}
double getX(double a) {
double x0 = a/5;
while(mabs(x: a - x0 * x0 * x0 * x0 * x0) >= 0.001){
x0 = 0.2 * (4 * x0 + a / (x0 * x0 * x0 * x0));
}
return x0;
}
Run: untitled23
/Users/marinaromanchukevich/CLionProjects/untitled23/cmake-build-debug/untitled23
122
2.613801
Process finished with exit code 0


onlinegdb.com
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
Run
Debug
Stop
Share
Save
Beautify
Language
main.S
Эквивалентное представление переменных в программе на C
В main:
rbp[-24] -- `double a`
rbp[-12] -- `int b`
rbp[-40] -- `a` (параметр в mabs)
rbp[-8] -- `double x`
В mabs:
rbp[-8] -- `double x` -- формальный параметр
В L9:
rbp[-40] -- `x`
В getX:
rbp[-24] -- `double a`
rbp[-8] -- `double x0`
.intel_syntax noprefix # Синтаксис в стиле Intel
.text # Начало секции
.globl mabs
.type mabs, @function
mabs: # Функция mabs
push rbp # / Сохраняем rbp на стек
mov rbp, rsp # | Вместо rbp записали rsp (rbp := rsp)
movsd QWORD PTR -8[rbp], xmm0 # `double x`
pxor xmm0, xmm0 # обнуление xmm0
ucomisd xmm0, QWORD PTR -8[rbp] # сравнение 0 и `x`
jbe .L7 # если 0 меньше или равен, переход к .L7
movsd xmm1, QWORD PTR -8[rbp] # xmm1 := `x`
movq xmm0, QWORD PTR .LC1[rip] # копирование 64 бита .LC1 в xmm0
xorpd xmm0, xmm1 # xmm0 ^= xmm1 (теперь xmm0 = -x)
input
122
2.613801
...Program finished with exit code 0
Press ENTER to exit console.
GOT AN OPINION?
SHARE AND GET REWARDED
Rakuten AIP
Have fun taking surveys and get paid!
ADS VIA CARBON
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online

Тест 5: -32



Выполнено в соответствии с критериями на 5 баллов

5. • Приведено решение задачи на C на планируемую оценку. Ввод данных осуществляется с клавиатры. Вывод данных осуществляется на дисплей. Файл idz3.c на гитхаб
6. • В полученную ассемблерную программу, откомпилированную без оптимизирующих и отладочных опций, добавлены комментарии, поясняющие эквивалентное представление переменных в программе на C. То есть, для всех

ссылок на память, включая и относи- тельные адреса и регистры, указать имя переменной на языке С исходной программы.

7. • Из ассемблерной программы убраны лишние макросы за счет использования при компиляции из С соответствующих аргументов командной строки и/или за счет ручного редактирования исходного текста ассемблерной программы.
   gcc -masm=intel -fno-asynchronous-unwind-tables -fno-jump-tables -fno-stack-protector -fno-exceptions -S 8.c
8. • Модифицированная ассемблерная программа отдельно откомпилирована и скомпонована без использования опций отладки.
9. • Представлено полное тестовое покрытие, дающее одинаковый результат на обоих программах. Приведены результаты тестовых прогонов для обоих программ, демонстрирующие эквивалентность функционирования.
   • Для сопоставления с полученной ассемблерной программой необходимо также приложить исходные тексты на ассемблере, сформированные компилятором языка С. Файл idz3noComm.s на гитхаб
10. • Сформирован отчет с результатами тестовых прогонов и описанием используемых опций компиляции, проведенных модификаций ассемблерной программы.
11. • В программе на языке С необходимо использовать функции с передачей данных через формальные параметры.
    Функция getX и mabs
12. • Внутри функций необходимо использовать локальные переменные, которые при компиляции отображаются на стек.
    Функция getX
13. • В ассемблерную программу в местах вызова функции добавить комментарии, описывающие передачу фактических параметров и перенос возвращаемого результата. При этом небходимо отметить, какая переменная или результат какого выражения соответствует тому или иному фактическому параметру.
    Файл idz3.s на гитхаб
14. • В ассемблерных функциях для каждого формального параметра необходимо добавить комментарии, описывающие связь между именами формальных параметров на языке С и регистрами (или значением на стеке), через которые эти параметры передаются.
15. • Для сопоставления с полученной ассемблерной программой необходимо также приложить исходные тексты на ассемблере, сформированные компилятором языка С.
16. • Информацию о проведенных изменениях отобразить в отчете наряду с информацией, необходимой на предыдущую оценку.