 **Бібліотеки <complex.h> та <tgmath.h>**

Виконала студентка другого курсу

механіко-математичного факультету

з групи комп’ютерна математика 1

Олександра Мікушова

|  |  |
| --- | --- |
| Комплексні числа на с та с++ |  |

Хедери <complex.h> і <tgmath.h> — це дві бібліотеки C, які забезпечують роботу з комплексними числами.

Бібліотека <complex.h> — хедер C, який підтримує роботу з комплексними числами. Він був написаний раніше за <tgmath.h>. Він визначає клас шаблону в стандартному просторі імен std, який дозволяє працювати з комплексними числами у коді C. Клас шаблону для роботи з комплексними числами надає низку функцій-членів і перевантажених операторів, які дозволяють виконувати типові математичні операції з комплексними числами, наприклад додавання, віднімання, множення та ділення.

Заголовок <tgmath.h> — це сучасніший заголовок C++, який підтримує роботу з будь-якими числами(з комплексними також). Він визначає низку типових макросів, які, по суті, є скороченими позначеннями для загальних математичних функцій. Це дозволяє писати код, який є більш гнучким і легшим для читання.

Функції <complex.h>

Мова програмування С підтримує три вбудовані типи змінних до яких можна отримати доступ через назви double \_Complex, float \_Complex  та long double \_Complex. Якщо у програму було включено хедер <complex.h>, змінні також можна оголошувати як double complex, float complex, long double complex.

Крім комплексного типу, в C також доступний уявний тип double \_Imaginary, float \_Imaginary та long double Imaginary (double Imaginary, float imaginery, long double imaginary якщо включено <complex.h>).

В залежності від типу змінної, в назву функцій в <complex.h> додають літеру (f (float) або l (long double)).

Функції:

CMPLX, CMPLXF,CMPLXL; створює комплексне число з дійсної та уявної частини

creal, crealf, creall; обчислює дійсну частину комплексного числа

cimag,cimagf, cimagl; обчислює уявну частину комплексного числа

cabs, cabsf, cabsl; обчислює модуль числа комплексного числа

carg, cargf, cargl; обчислює аргумент

conj, conjf, conjl; обчислює суміжне число

cproj, cprojf, cprojl; обчислює проекцію на сферу Рімана

cexp, cexpf, cexpl; обчислює експоненту

clog, clogf, clogl; обчислює натуральний логарифм

cpow, cpowf, cpowl; підносить до степеня

csqrt, csrtf, csqrtl; обчислює квадратний корінь

csin, csinf, csinl; обчислює синус

ccos, ccosf, ccosl; обчислює косинус

ctan, ctanf, ctanl; обчислює тангенс

casin, casinf, casinl; обчислює арксинус

cacos, cacosf, cacosl; обчислює арккосинус

catan, catanf, catanl; обчислює арктангенс

csinh, csinhf, csinhl; обчислює гіперболічний синус

cacosh, cacoshf, cacoshl; обчислює гіперболічний косинус

catanh, catanhf, catanhl; обчислює гіперболічний тангенс

Функції <tgmath.h>

<tgmath.h> визначає набір загальних математичних функцій, які працюють з різними типами даних. Ці функції можна використовувати для виконання звичайних математичних операцій над числами з плаваючою комою та комплексними числами, а також іншими типами чисел, які підтримуються стандартною бібліотекою.

Функції в <tgmath.h> оголошуються за допомогою загального типу \_\_T, який є змінною типу, яка представляє тип аргументів, що передаються функції. Фактичний тип, який використовується для виклику певної функції, визначається під час компіляції зважаючи на тип аргументів, переданих функції. Це дозволяє використовувати одну функцію з різними типами даних без необхідності використовувати явні приведення типів або писати окремі функції для кожного типу. Відмінність <tgmath.h> від <complex.h> полягає у тому, що <tgmath.h> працює з усіма типами змінних, а <complex.h> лише з комплексними.

Функції:

Тригонометричні: sin,cos, tan, asin, acos, atan, sinh, cosh, tanh,asinh, acosh, atanh.

сbrt корінь кубчний

ceil обчислює найменше ціле число не менше за аргумент

copysign(x,y) величина з плаваючою комою з величиною x і знаком y

erf обчислює значення функції помилок Гауса

erfc обчислює значення комплементарної функції помилок

exp2 значення 2 в степені заданого числа

expm1 значення числа Ейлера в заданому степені

fdim позитивна різниця двох значень

floor найбільше ціле число не більше за.аргумент

fmax максимальне з. двох значень

fmin мінімальне з двох значень

fmod(x,y) остача від ділення x/y

frexp Розкладає задане значення з плаваючою комою на нормалізований дріб і інтегральний ступінь двійки

ldexp(x,y) Множить значення з плаваючою комою x на число 2, зведене до степеня y

lgamma натуральний логарифм від значення гамма функції

lround обчислює найближче ціле значення до arg (у форматі з плаваючою комою), незалежно від поточного режиму округлення

lrint обчислює аргумент аргументу з плаваючою комою до цілого значення у форматі з плаваючою комою, використовуючи поточний режим округлення

nextafter(from,to) Спочатку перетворює обидва аргументи на тип функції, а потім повертається наступне представлене значення from y напрямку до to. Якщо from дорівнює to, повертається to

remainder(x,y) остача IEEE від ділення x/y

rint округлює аргумент аргументу з плаваючою крапкою до цілого значення у форматі з плаваючою крапкою, використовуючи поточний режим округлення

round найближче ціле число

tgamma обчислює гамма-функцію від заданого числа

trunc найближче ціле число, не більше за задане

Бібліотека <complex>

<complex> — це шаблонний клас у стандартній бібліотеці C++, який визначає набір функцій для роботи з комплексними числами. Комплексне можна представити у формі a + bi, де a і b — дійсні числа, а i — уявна одиниця. Клас шаблону <complex> дозволяє працювати з комплексними числами програмах на C++, а також надає ряд корисних функцій для роботи з комплексними числами.

Оператори:

operator+ сума комплесних чисел

operator- різниця комплексних чисел

operator\* добуток комплесних чисел

operator/ частка комплесних чисел

operator==

operator!= порівняння двох комплексних чисел або комплексного та скаляра

operator>> серіалізує комплексне число

operator<< десеріалізує

Функції:

real дійсна частина

imag уявна частина

abs модуль

arg кут

norm модуль в квадраті

conj спряжене комлексне число

proj проекція на сферу Рімана

polar створює комплексне число з кута та модуля

exp експонента

log логарифм натуральний від комплексного числа

log10 логарифм за основою 10 від комплексного числа

pow піднесення до степеня

sqrt корінь квадратний

Тригонометричні: sin,cos, tan, asin, acos, atan, sinh, cosh, tanh,asinh, acosh, atanh.

Результати запусків функцій

class MyComplex <complex> <complex.h><tgmath.h>

**kvadratne**() **a**(3,4),**b**(5,2),**c**(7,8)

x1 -0.494611+1.68996i (-0.494611,1.68996) -0.494611+1.689964i

x2 -0.425389-1.12996i (-0.425389,-1.12996) -0.425389+-1.129964i

Elapsed time: 0.000195 seconds 0.000236 seconds 0.000142

**kubichne**() **a**(3,4),**b**(5,2),**c**(7,8), **d**(8,9)

x1 1.17694-1.02397i (1.17694,-1.02397) 1.176935+-1.023966i

x2 -1.53231+0.396879i (-1.53231,0.396879) -1.532307+0.396879i

x3 0.355372+0.627087i (0.355372,0.627087) -1.793422+0.842821i

Elapsed time: 0.000043 seconds 0.000049 seconds 0.000059

**chetvertoho**() **a**(2,0),**b**(1,0),**c**(-11,0), **d**(1,0), **e**(2,0)

x1=1.54342+1.13002i (1.54342,1.13002) 1.543422+1.130017i

x2=-1.79342+0.842821i (-1.79342,0.842821) -1.793422+0.842821i

x3=1.54342-1.13002i (1.54342,-1.13002) 1.543422+-1.130017i

x4=-1.79342-0.842821i (-1.79342,-0.842821) 1.793422+-0.842821i

Elapsed time: 0.000057 seconds 0.000041 0.000035

У моєму проекті було порівняно результати та швидкодію функцій на С (з використанням <complex.h> та <tgmath.h>), С++ (з використанням <complex>) та C++ (з класом MyComplex). Аналізуючи результати, я приходжу до висновку, що функції kvadratne() та chetvertoho(), що обчислюють відповідно розв’язки квадратного рівняння та рівняння четвертого степеня найшвидше працюють на С, функція kubichne() найшвидше обчислює розв’язки кубічних рівнянь з класом MyComplex на С++.