****

图书馆代码系列教程

|  |  |
| --- | --- |
| 题目: | 任意底的对数 |
| 类型: | 高级代码(HC) |

|  |  |
| --- | --- |
| 作者: | Xc136 |
| 图书馆编号: |  |
| 所属组(选填): | 红与白 |
| 完成日期: | 2025.06.06 |
| 基于版本: | 1.16 |
| 联系方式（选填）: |  |

摘 要

这里提供了一种可以开任意底的对数的方式

**关键词：高级代码；钢铁雄心 4 教程；对数计算；变量计算**

这里需要封装

calculate\_ln = {

然后在里面设置好常数（缩放因子越大精度越高）

set\_temp\_variable = { ln2 = 6931 }        # ln(2) ≈ 0.6931 \* 10000

set\_temp\_variable = { ln10 = 23026 }      # ln(10) ≈ 2.3026 \* 10000

set\_temp\_variable = { scale\_factor = 10000 } # 缩放因子

将要用的变量设定好（这里要从外部设定input\_value）

set\_temp\_variable = { current = input\_value }

    set\_temp\_variable = { exponent = 0 }

    set\_temp\_variable = { result = 0 }

处理不在【2到0.5】之间的数字（这里if次数越多，能输入的数越大8次能输入小于512的）

if = {

        limit = { check\_variable = { current > 2 } }

        divide\_temp\_variable = { current = 2 }

        add\_to\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current > 2 } }

        divide\_temp\_variable = { current = 2 }

        add\_to\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current > 2 } }

        divide\_temp\_variable = { current = 2 }

        add\_to\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current > 2 } }

        divide\_temp\_variable = { current = 2 }

        add\_to\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current > 2 } }

        divide\_temp\_variable = { current = 2 }

        add\_to\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current > 2 } }

        divide\_temp\_variable = { current = 2 }

        add\_to\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current > 2 } }

        divide\_temp\_variable = { current = 2 }

        add\_to\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current > 2 } }

        divide\_temp\_variable = { current = 2 }

        add\_to\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current > 2 } }

        divide\_temp\_variable = { current = 2 }

        add\_to\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current > 2 } }

        divide\_temp\_variable = { current = 2 }

        add\_to\_temp\_variable = { exponent = 1 }

    }

    # 处理小于1的值

    if = {

        limit = { check\_variable = { current < 0.5 } }

        multiply\_temp\_variable = { current = 2 }

        subtract\_from\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current < 0.5 } }

        multiply\_temp\_variable = { current = 2 }

        subtract\_from\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current < 0.5 } }

        multiply\_temp\_variable = { current = 2 }

        subtract\_from\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current < 0.5 } }

        multiply\_temp\_variable = { current = 2 }

        subtract\_from\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current < 0.5 } }

        multiply\_temp\_variable = { current = 2 }

        subtract\_from\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current < 0.5 } }

        multiply\_temp\_variable = { current = 2 }

        subtract\_from\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current < 0.5 } }

        multiply\_temp\_variable = { current = 2 }

        subtract\_from\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current < 0.5 } }

        multiply\_temp\_variable = { current = 2 }

        subtract\_from\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current < 0.5 } }

        multiply\_temp\_variable = { current = 2 }

        subtract\_from\_temp\_variable = { exponent = 1 }

    }

    if = {

        limit = { check\_variable = { current < 0.5 } }

        multiply\_temp\_variable = { current = 2 }

        subtract\_from\_temp\_variable = { exponent = 1 }

    }

设定一下x值

set\_temp\_variable = { x = current }

    subtract\_from\_temp\_variable = { x = 1 }

利用泰勒展开得到了

set\_temp\_variable = { x2 = x }

    multiply\_temp\_variable = { x2 = x }

    # 计算 x³

    set\_temp\_variable = { x3 = x2 }

    multiply\_temp\_variable = { x3 = x }

    # 计算 x⁴

    set\_temp\_variable = { x4 = x3 }

    multiply\_temp\_variable = { x4 = x }

    set\_temp\_variable = { x5 = x4 }

    multiply\_temp\_variable = { x5 = x }

    set\_temp\_variable = { x6 = x5 }

    multiply\_temp\_variable = { x6 = x }

    set\_temp\_variable = { x7 = x6 }

    multiply\_temp\_variable = { x7 = x }

    set\_temp\_variable = { x8 = x7 }

    multiply\_temp\_variable = { x8 = x }

    # 计算各项

    set\_temp\_variable = { term1 = x }

    set\_temp\_variable = { term2 = x2 }

    divide\_temp\_variable = { term2 = 2 }

    set\_temp\_variable = { term3 = x3 }

    divide\_temp\_variable = { term3 = 3 }

    set\_temp\_variable = { term4 = x4 }

    divide\_temp\_variable = { term4 = 4 }

    set\_temp\_variable = { term5 = x5 }

    divide\_temp\_variable = { term5 = 5 }

    set\_temp\_variable = { term6 = x6 }

    divide\_temp\_variable = { term6 = 6 }

    set\_temp\_variable = { term7 = x7 }

    divide\_temp\_variable = { term7 = 7 }

    set\_temp\_variable = { term8 = x8 }

    divide\_temp\_variable = { term8 = 8 }

    # 组合各项

    set\_temp\_variable = { result = term1 }

    subtract\_from\_temp\_variable = { result = term2 }

    add\_to\_temp\_variable = { result = term3 }

    subtract\_from\_temp\_variable = { result = term4 }

    add\_to\_temp\_variable = { result = term5 }

    subtract\_from\_temp\_variable = { result = term6 }

    add\_to\_temp\_variable = { result = term7 }

    subtract\_from\_temp\_variable = { result = term8 }

这个答案就是只有一部分的结果

要加上前面计数的指数

multiply\_temp\_variable = { result = scale\_factor }

    set\_temp\_avriable = { exp\_part = exponent }

    multiply\_temp\_variable = { exponent = ln2 }

    add\_to\_temp\_variable = { result = exponent }

最后把缩放因子去掉

divide\_temp\_variable = { result = scale\_factor }

这样就得到了一个可以开以e为底的对数的代码

接着利用logx/y=lnx/lny

calculate\_log\_base = {

    # 计算ln(input\_value)

    set\_temp\_variable = { save\_base = base } # 保存base

    set\_temp\_variable = { input\_value = input\_value\_log\_base }

    calculate\_ln = yes

    set\_temp\_variable = { ln\_input = result }

    # 计算ln(base)

    set\_temp\_variable = { input\_value = save\_base }

    calculate\_ln = yes

    set\_temp\_variable = { ln\_base = result }

    # 计算 log\_base(input\_value) = ln(input\_value) / ln(base)

    set\_temp\_variable = { result\_log\_base = ln\_input }

    divide\_temp\_variable = { result\_log\_base = ln\_base }

}

将x与y代如刚刚的封装里，就可以计算任意底的对数了