

P1163 银行贷款

题目描述

当一个人从银行贷款后，在一段时间内他(她)将不得不每月偿还固定的分期付款。这个问题要求计算出贷款者向银行支付的利率。假设利率按月累计。

输入格式

三个用空格隔开的正整数。

第一个整数表示贷款的原值，第二个整数表示每月支付的分期付款金额，第三个整数表示分期付款还清贷款所需的总月数。

输出格式

一个实数，表示该贷款的月利率(用百分数表示)，四舍五入精确到 0.1%。

输入样例

1000 100 12

输出样例

2.9

解析

核心思想是利用二分法来枚举利率。

在这里存在一个金融知识：月利率。那么什么是月利率呢？举个例子：

对于第 1 个月，他欠银行 $base + base * x$ ，但由题意得，他每个月向银行还 m 元钱，所以下个月前还欠银行 $base + base * x - m$ ；来到第 2 个月，欠银行 $(base + base * x - m) + (base + base * x - m) * x$ ，还款 m 元，以此类推，直到第 k 个月后，求出他还欠银行的钱数；如果欠的钱数 ≤ 0 ，说明在月利率为 x 的情况下，每月还 m 元是可以还清的，返回 `true`，反之，返回 `false`。

抽象成代码就是：

```
bool check(double x) {  
    double base = n;  
    for (int i = 0; i < k; ++i) {  
        //当月欠的钱  
        base *= (1 + x / 100);  
        //当月还的钱  
        base -= m;  
    }  
}
```

```

        //在当前利率下可以还清贷款
        if (base <= 0) {
            return true;
        }

        //当前利率还不清
        else {
            return false;
        }
    }
}

```

另外一个难点是二分的左右区间，左区间很容易想到是0，那么右区间呢？题目中没有给出明确的上限，因此我们不妨把区间开大一些，防止出错。老师在这里设置的区间范围为0-1000。

最后一个需要注意的就是二分的终止条件，和求浮点数的函数方程解一样，我们需要对浮点数进行精度的比较，即定义eps变量。本题要求保留百分数的1位小数，那么总共就是3位小数，因此我们的eps变量应该需要再多一位，即 10^{-4}

编码

```

#include <bits/stdc++.h>

using namespace std;

double n, m, k;

const double eps = 1e-4; //定义我们计算的精度

//计算k个月需要缴纳的贷款
bool check(double x) {
    double base = n;
    for (int i = 0; i < k; ++i) {
        //当月欠的钱
        base *= (1 + x / 100);
        //当月还的钱
        base -= m;
    }

    //在当前利率下可以还清贷款
    if (base <= 0) {
        return true;
    }

    //当前利率还不清
    else {
        return false;
    }
}

//执行二分搜索
double BinarySearch(double l, double r) {
    double mid;

```

```

//通过与eps作比较来判断是否找到根
while (r-l > eps) {
    //计算中值
    mid = l + (r - l) / 2;
    bool res = check(mid);
    //能够还清, 尝试提高利率
    if (res) {
        l = mid;
    }
    //不能还清, 降低利率
    else {
        r = mid;
    }
}
return l;
}

int main(int argc, char *argv[]) {
    cin >> n >> m >> k;
    double l = 0, r = 1000;
    double res = BinarySearch(l, r);
    printf("%.11f ", res);
    return 0;
}

```

逻辑航线培优教育, 信息学奥赛培训专家。

扫码添加作者获取更多内容。

