**北京科技大学实验报告**

学院： 计通学院 专业：计算机科学与技术 班级： 计1503

姓名： 唐誉源 学号：41503302 实验日期： 2017 年 11 月 16 日

#### **实验名称：**

算法设计与分析实验1分治法

#### **实验目的：**

1. 利用分治思想求解大数n次方的模运算
2. 要求多组测试数据输入输出
3. 控制分步计算结果不能超出数据类型的范围

#### **实验设备（环境）及要求：**

操作系统: macOS High Sierra 10.12.1;

编辑器: sublime txt 3;

编译链接: gcc.

#### **实验原理：**

1. 模运算乘法分配律：
2. 分治思想：，在为偶数的情况下，等价于；在为奇数的情况下，等价于.
3. 将为了不直接计算导致中间结果超出int表示范围，利用模运算乘法分配律来改写(2)中的公式：Z为偶数的情况下：；在Z为奇数的情况下：

#### **实验内容与步骤：**

##### *（1）实验内容*

计算，其中和的范围是[0, 2147483647]，的范围是[1, 46340]。要求输入多组测试数据，每组测试数据占一行，三个整数Y, Z, D；输出X，每组输出占一行。

##### *（2）主要步骤*

由实验原理中的结合分治思想和模运算乘法分配律所得到的公式编写计算$X=Y^Z\bmod D$的函数：

int count(int Y, int Z, int D) {  
 **if** (Z == 1) {  
 **return** Y % D;  
 } **else** **if** (Z%2 == 0) {  
 **return** count(Y, Z/2, D) \* count(Y, Z/2, D) % D; *// (Y\*Y)^(Z/2) mod D*  
 } **else** {  
 **return** (count(Y, Z/2, D) \* count(Y, Z/2, D) % D) \*(Y % D) % D;*// (Y\*Y)^(Z/2)\*Y mod D*  
 }  
}

在设计main函数时，为了保证多组测试数据的输入，设置成每次读入一行的三个数分别作为Y、Z和D，之后直接计算出其结果X记录到数组中。循环上述步骤，并在遇到结束符（在此设置为'0 0 0'）时停止读入，开始输出数组X中记录的结果。

int main() {  
 int Y, Z, D;  
 int i,j;  
 int X[100];  
 printf("Please input Y, Z, D (Y and Z belongs to [0, 2147483647] and D belongs to [1, 46340]), end with '0 0 0': \n");  
 scanf("%d %d %d", &Y, &Z, &D);  
 **for** (i = 0; Y; ++i) {  
 X[i] = count(Y, Z, D);  
 scanf("%d %d %d", &Y, &Z, &D);  
 }  
 printf("X:\n");  
 **for** (j = 0; j < i; ++j)  
 {  
 printf("%d\n", X[j]);  
 }  
 **return** 0;  
}

#### **实验数据：**

样例输入：

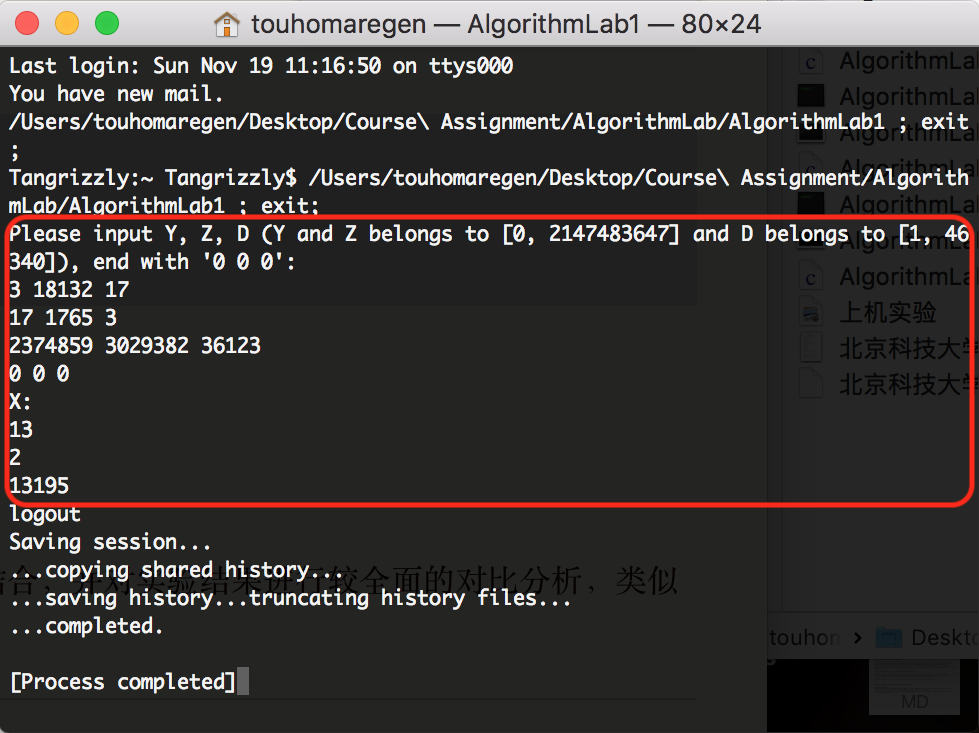
3 18132 17  
17 1765 3  
2374859 3029382 36123

样例输出：

13  
2  
13195

#### **实验结果与分析：**

运行结果截图展示：



可以看出运行结果完整而正确地显示了计算结果。利用分治法逐步化简问题，并且经过模运算乘法分配律的处理，保证了计算的中间结构不超过int的表示范围，从而得到正确的结果。

#### **实验结论：**

##### *（1）实验结论*

利用分治思想将大数n次方的模运算的问题分类成幂为奇数和幂为偶数的问题，逐步简化运算。利用模运算乘法分配律简化问题，确保计算中间结果逐渐变小，不超过数据类型表示范围。在main函数中利用设定结束符和数组保存结果的方法实现多组测试数据的输入输出。

##### *（2）讨论*

1. 输入数据Y和Z的大小是int非负数表示范围，在本程序中直接利用int来记录其数据和运算中间结果，并用模运算乘法分配律保证范围。如果不利用模运算乘法分配律来改变计算公式，可以直接用long类型来表示。这样的方法简化了思维过程，但提高了存储单元成本；
2. 实现多组测试数据的输入输出的问题时，也可以事先提前输入测试数据的组数，以替代结束符的应用；
3. 选择输入数据之后直接调用函数计算结果记录在数组中，而不是用数组先记录输入的测试数据之后在进行函数调用输出结果的方法，减少了存储单元的使用量；
4. 测试数据组数被控制在了100以下，如果超出范围需要调整程序中记录运算结果X的数组大小；
5. 程序由gcc直接编译得到Unix executable格式文件，在window上运行需要重新编译连接得到.exe文件。