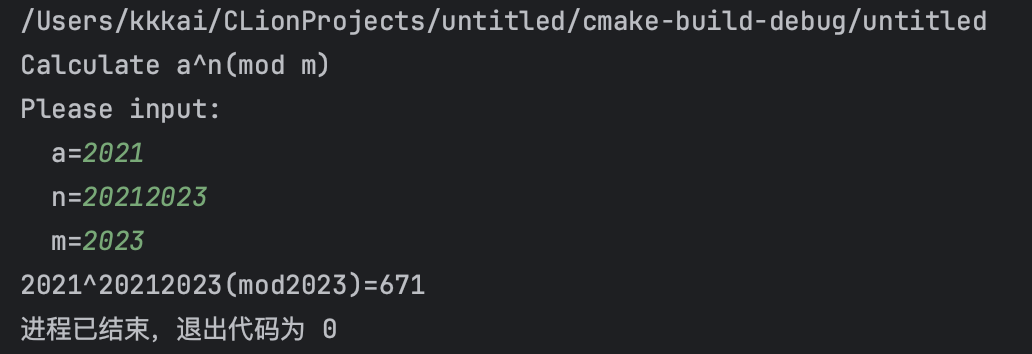
# 第2次编程练习报告

姓名：申宗尚 学号：2213924 班级：信安班

##### 编程练习1——平方-乘算法

* **源码部分：**
* #include <iostream>  
  using namespace std;  
    
  long long func(long long base, long long exp, long long mod) {  
   long long result = 1;  
   while (exp > 0) {  
   if (exp % 2 == 1) {  
   result = (result \* base) % mod;  
   }  
   base = (base \* base) % mod;  
   exp = exp / 2;  
   }  
   return result;  
  }  
    
  int main() {  
   cout << "Calculate a^n(mod m)" << endl;  
   cout << "Please input:" << endl;  
   int base, exp, mod;  
   cout << " a=";  
   cin >> base;  
   cout << " n=";  
   cin >> exp;  
   cout << " m=";  
   cin >> mod;  
   long long result = func(base, exp, mod);  
   cout << base << "^" << exp << "(mod" << mod << ")=" << result;  
   system("pause");  
   return 0;  
  }
* **说明部分：通过base，exp，mod记录输入的底数，指数和模数，再通过func函数，进行求解。在func函数中，利用了幂次的二进制展开进行快速计算，减少乘法和取模的次数，提高了计算效率。具体如下：将指数exp用二进制表示，然后根据其每一位是否为 1 决定是否乘以a，然后递归地利用a^2^k的性质计算，这样可以在O（logn）的时间复杂度内完成幂运算。**
* **运行示例：**



##### 编程练习2——扩展的欧几里得算法求逆元

**源码部分：**

#include <iostream>  
using namespace std;  
int gcd(int a, int b) {  
 while (b != 0) {  
 int temp = b;  
 b = a % b;  
 a = temp;  
 }  
 return a;  
}  
int lcm(int a, int b) {  
 return (a \* b) / gcd(a, b);  
}  
// 扩展欧几里得算法，求模逆元  
void extendedEuclidean(int a, int b, int &x, int &y) {  
 if (b == 0) {  
 x = 1;  
 y = 0;  
 return;  
 }  
 int x1, y1;  
 extendedEuclidean(b, a % b, x1, y1);  
 x = y1;  
 y = x1 - (a / b) \* y1;  
}  
int main() {  
 int a, b;  
 cout << "a=";  
 cin >> a;  
 cout << "b=";  
 cin >> b;  
 int a\_inv, b\_inv;  
 extendedEuclidean(a, b, a\_inv, b\_inv);  
 a\_inv = (a\_inv % b + b) % b;  
 b\_inv = (b\_inv % a + a) % a;  
 cout << "gcd(a,b)=" << gcd(a,b);  
 cout << endl << "lcm(a,b)=" << lcm(a,b);  
 cout << endl << "a^(-1)=" << a\_inv << "(mod " << b << ")";  
 cout << endl << "b^(-1)=" << b\_inv << "(mod " << a << ")";  
 return 0;  
}

**说明部分:** **输入两个整数a和b后，用gcd和lcm函数分别计算它们的最大公约数和最小公倍数。然后调用extendedEuclidean函数来计算a和b的模逆元。**

**在扩展欧几里得算法中，要求解两个整数a和b的最大公约数，同时计算出ax + by = gcd(a, b) 的整数x和y。这里使用递归的方法实现算法，基本思想是利用等式的递归性质：**

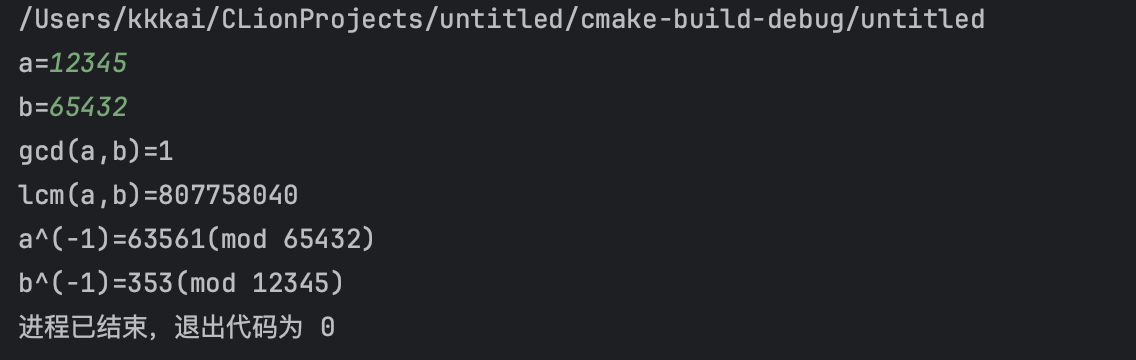
**如果b等于0，则gcd(a, 0) = a，同时x = 1，y = 0。**

**否则，我们递归地计算b和a % b的最大公约数，并且同时求得x1和y1，然后可以得到x = y1，y = x1 - (a / b) \* y1。**

**这样，我们就可以用递归的方式求得了x和y，**

**同时在主函数中还需要注意，模逆元的计算要确保其为正数。因此，对求得的模逆元a\_inv和b\_inv进行了修正，保证其为正数。**

**运行示例：**

****