第四次编程练习报告

姓名：张洋　　学号：2111460　　班级：信安二班

编程练习 1——编程实现求解最小原根并基于最小原根构造指数表

```
#include<iostream>
#include<cmath>
#include<iomanip>
using namespace std;

int gcd(int a, int b) { //辗转相除法求最大公因子
    if (a < b) //a为大的那个数
        swap(a, b);
    int r = a % b; //余数
    while (r != 0) { //当余数不为0时
        a = b;
        b = r;
        r = a % b;
    }
    return b;
}

int reduce_system(int* r, int n) {//求缩系 1,2,3,4,5,6,7.....
    int count = 0;//与n互素的元素个数
    for (int i = 1; i < n; i++) {
        if (gcd(i, n) == 1) {
            r[count] = i;
            count++;
        }
    }
    return count;
}

int deposition(int n, int* d) { //求素因子分解 2*3*17
    int j = 0;//记录因子个数
    for (int i = 2; i <= n; i++) {
        int count = 0; //count次幂
        while (n % i == 0) { //如果能整除
            count++; //次数+1
            n /= i; //除掉因子
        }
```

```
        if (count != 0) { //结果存入d
            d[j] = pow(i, count);
            j++;
        }
    }
    return j;
}

int remainder(int a, int n, int m) {//求a^n mod m，避免溢出
    int r = a % m;
    for (int i = 1; i < n; i++)
        r = (r * a) % m;
    return r;
}

int primary_root(int count, int n,int *r) {//求最小原根
    int* d = new int[n - 1];//记录互素的因子
    int count_deposition = deposition(count, d);
    int* divide = new int[count_deposition];//用count分别除以d中的因子
    for (int i = 0; i < count_deposition; i++)
        divide[i] = count / d[i];
    for (int i = 1; i < count; i++) {
        bool flag = true;
        for (int j = 0; j < count_deposition; j++) {
            if (remainder(r[i], divide[j], n) == 1) {
                flag = false;
                break;
            }
        }
        if (flag == true)
            return r[i];
    }
}

void index_table(int n, int count, int root) {//构建指数表
    int** table = new int* [n / 10 + 1];
    for (int i = 0; i < n / 10 + 1; i++)
        table[i] = new int[10];
    for (int i = 0; i < n / 10 + 1; i++) {//初始化table中的值都为-1
        for (int j = 0; j < 10; j++) {
            table[i][j] = -1;
        }
    }
    table[0][1] = 0;
```

```
    for (int i = 1; i < count; i++) { //把指数加入到表中
        int r = remainder(root, i, n);
        table[r / 10][r % 10] = i;
    }
    cout << setw(5) << " ";//输出
    for (int i = 0; i < 10; i++)
        cout << setw(5) << i;
    cout << endl;
    for (int i = 0; i < n / 10 + 1; i++) {
        cout << setw(5) << i;
        for (int j = 0; j < 10; j++) {
            if (table[i][j] == -1)
                cout << setw(5) << "-";
            else
                cout << setw(5) << table[i][j];
        }
        cout << endl;
    }
}

int main() {
    int n;
    cout << "Please input n(n>0):";
    cin >> n;
    int* r = new int[n-1];//存放模n-1的缩系
    int len = reduce_system(r, n);
    int* result = new int[len];//存放所有原根
    int root = primary_root(len, n, r);
    cout << "The min primitive root of " << n << ":g=" << root << endl;
    cout << "The ind_table of " << n << " based on g=" << root << " is:" << endl;
    index_table(n, len, root);
}
```

说明部分：

由书中定理 4.2.12，设 m 是大于 2 的整数，φ(m)的所有不同的素因子是$q_1, q_2, \ldots, q_s$，则与 m 互素的正整数 g 是 m 的一个原根的充要条件是

$$g^{\frac{\varphi(m)}{q_i}} \not\equiv 1 \pmod{m}, \quad i = 1, 2, \cdots, s.$$

根据此定理可以得出最小原根，进而可以根据最小原根构造指数表。

运行示例：

```
Please input n(n>0):103
The min primitive root of 103: g=5
The ind_table of 103 based on g=5 is:
         0    1    2    3    4    5    6    7    8    9
    0    -    0   44   39   88    1   83    4   30   78
    1   45   61   25   72   48   40   74   70   20   80
    2   89   43    3   24   69    2   14   15   92   86
    3   84   57   16  100   12    5   64   93   22    9
    4   31   50   87   77   47   79   68   85   11    8
    5   46    7   58   97   59   62   34   17   28   98
    6   26   36  101   82   60   73   42   13   56   63
    7   49   67    6   33   35   41   66   65   53   18
    8   75   54   94   38   29   71   19   23   91   99
    9   21   76   10   96   27   81   55   32   52   37
   10   90   95   51    -    -    -    -    -    -    -
```

```
Please input n(n>0):169
The min primitive root of 169: g=2
The ind_table of 169 based on g=2 is:
         0    1    2    3    4    5    6    7    8    9
    0    -    0    1  124    2    9  125  107    3   92
    1   10  103  126    -  108  133    4  146   93   65
    2   11   75  104  130  127   18    -   60  109   40
    3  134   21    5   71  147  116   94  151   66    -
    4   12   85   76  122  105  101  131   63  128   58
    5   19  114    -  120   61  112  110   33   41   35
    6  135  140   22   43    6    -   72   37  148   98
    7  117  137   95   51  152  142   67   54    -   24
    8   13   28   86   45   77  155  123    8  106   91
    9  102    -  132  145   64   74  129   17   59   39
   10   20   70  115  150    -   84  121  100   62   57
   11  113  119  111   32   34  139   42    -   36   97
   12  136   50  141   53   23   27   44  154    7   90
   13    -  144   73   16   38   69  149   83   99   56
   14  118   31  138    -   96   49   52   26  153   89
   15  143   15   68   82   55   30    -   48   25   88
   16   14   81   29   47   87   80   46   79   78    -
```