# 第 4 次编程练习报告

姓名：武桐西　　学号：2112515　　班级：信安一班

## 一、编程练习 1——求解最小原根并基于最小原根构造指数表

### ➢ 源码部分：

```
#include<iostream>
#include<vector>
#include<math.h>
using namespace std;
//求解m的最小原根g并基于最小原根构造指数表

bool isCoprime(int a, int b) {
	//辗转相除法求最大公因数，判断a,b是否互素
	a = a > 0 ? a : -a;//转为正数
	b = b > 0 ? b : -b;//转为正数
	if (a < b) {//令a为较大者，b为较小者
		int tmp = a;
		a = b;
		b = tmp;
	}
	while (a % b) {//余数不为0，继续循环
		int r = a % b;
		a = b;
		b = r;
	}
	return b == 1;
}

int phi(int m) {//求解\phi(m)
	int count = 0;//记录m的最小非负完全剩余系中与m互素的整数个数
	for (int i = 1; i <= m; i++) {
		if (isCoprime(m, i))
			count++;
	}
	return count;//phi(m)
}

void Eratosthenes(int n, bool*& A) {//Eratosthenes筛法
```

```
    //n为代求范围[2,n]，A保存整数表
    A = new bool[n + 1]{ false };
    for (int i = 2; i <= n; i++)
        A[i] = true;//初始化为true
    if (n < 4)
        return;//递归结束
    int n_ = (int)(sqrt(1.0 * n));
    bool* A_ = nullptr;
    Eratosthenes(n_, A_);
    for (int i = 2; i <= n_; i++) {
        if (A_[i]) {
            for (int j = 2; j <= n; j++) {
                if (j % i == 0 && j != i)
                    A[j] = false;
            }
        }
    }
}

void PrimeFactor(int n, vector<int>& Q) {
    //n = phi(m)
    //先用Eratosthenes筛法求[2,n]内所有素数
    bool* A = nullptr;
    Eratosthenes(n, A);
    int a = n;
    for (int i = 2; i <= n; i++) {
        if (A[i]) {//i为素数
            int count = 0;//记录素因子指数
            while (a % i == 0) {
                count++;
                a /= i;
            }
            if (count) {//count非零
                Q.push_back(n / i);
            }
        }
    }
}

int FMM(int a, int n, int m) {//平方-乘算法，快速模幂运算
    //return a^n(mod m)
    int N = int(log2(1.0 * n)) + 1;//用来计算n转为二进制时的位数
    bool* A = new bool[N];//保存n的二进制
    for (int i = N - 1; i >= 0; i--) {
```

```
        A[i] = bool(n % 2);//n转为二进制
        n /= 2;
    }
    int c = 1;//初始化c
    for (int i = 0; i < N; i++) {//从高位开始
        c *= c;
        c %= m;
        if (A[i])
            c = (c * a) % m;
    }
    return c;
}

bool MinPrimitiveRoot(int m, int& g) {
    //求m的最小原根g
    if (m == 1)//m==1, 不存在原根
        return false;
    if (m == 2) {//m为2，单独处理
        g = 1;
        return true;
    }
    vector<int> Q;
    PrimeFactor(phi(m), Q);//求phi(m)的素因子
    int i = 1;
    for (; i <= m; i++) {
        if (isCoprime(m, i)) {//m,i互素
            int len = Q.size();
            int j = 0;
            for (; j < len; j++) {
                if (FMM(i, Q[j], m) == 1) {
                    break;//失败
                }
            }
            if (j == len) {
                g = i;//最小原根
                return true;
            }
        }
    }
    return false;
}

void IndTable(int m) {
    int g;
```

```
if (MinPrimitiveRoot(m, g)) {//原根存在
    int phi_m = phi(m);
    int length = m / 10 + 1;//行数
    int** A = new int* [length];//存储Ind_Table
    for (int i = 0; i < length; i++) {
        A[i] = new int[10];
        for (int j = 0; j < 10; j++)
            A[i][j] = -1;//默认值为-1
    }
    A[0][1] = 0;
    for (int i = 1; i < phi_m; i++) {
        int a = FMM(g, i, m);
        A[a / 10][a % 10] = i;
    }
    //输出指数表
    cout << "The Min Primitive Root of " << m << ": g = " << g << endl;
    cout << "The Ind_Table of " << m << " base on g = " << g << " is:\n";
    cout << "\t";
    for (int i = 0; i < 10; i++) {
        cout << "\t" << i;
    }
    cout << endl;
    for (int i = 0; i < length; i++) {
        cout << "\t" << i;
        for (int j = 0; j < 10; j++) {
            cout << "\t";
            if (A[i][j] != -1) {
                cout << A[i][j];
            }
            else {
                cout << "-";
            }
        }
        cout << endl;
    }
    //回收内存
    for (int i = 0; i < length; i++) {
        delete[]A[i];
    }
    delete[]A;
}
else {
    cout << m << "的原根不存在!\n";
}
```

```
}


int main() {
    cout << "Please Input n (n>0): ";
    int m;
    cin >> m;
    IndTable(m);
    system("pause");
    return 0;
}
```

➢ **说明部分：**

1. 求解 m 的最小原根：

利用教材中定理 4.2.12，根据$g^{\frac{\varphi(m)}{q_i}}$模$m$与1不同余来判断$g$是否是$m$的原根（其中，$q_i$是$\varphi(m)$的不同的素因子）。

从$g$的最小正缩系中自小到大遍历，若当前的$g$满足定理的条件（对所有的$q_i$都有$g^{\frac{\varphi(m)}{q_i}}$模$m$与1不同余），则当前的$g$即为$m$的最小原根；若有一个素因子$q_j$使得$g^{\frac{\varphi(m)}{q_i}} \equiv 1(mod\ m)$，则当前的$g$不是$m$的原根。

若遍历完未找到，则说明$m$不存在原根。

注意：

① 当$m = 1,2$时，由于不满足定理 4.2.12 中的使用条件（$m > 2$），因此需要单独考虑。具体而言，当$m = 1$时，原根不存在；当$m = 2$时，最小原根为$g = 1$。

② 可以利用算术基本定理（在实现算术基本定理时，用到 Eratosthenes 筛法求素数）求解$\varphi(m)$的所有不同的素因子。

③ 在对$g$进行判断时，只需考虑与$m$互素的整数即可（$m$的最小

正缩系）。

④ 可以利用平方-乘算法，求解模幂运算$g^{\frac{\varphi(m)}{q_i}}\ (mod\ m)$。

⑤ $\varphi(m)$的值可以通过定义求解（最小正缩系中与$m$互素的整数个数）。

(2) 基于$m$的最小原根$g$，构造模$m$的指数表：

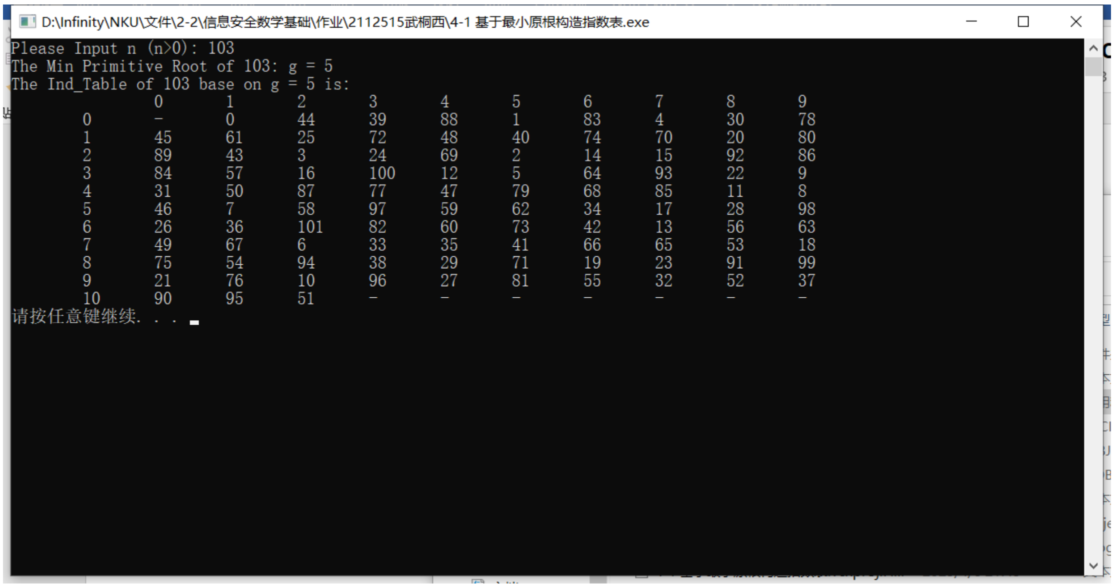
当$m$的原根存在时，根据$m$的最小原根$g$构造模$m$的指数表。

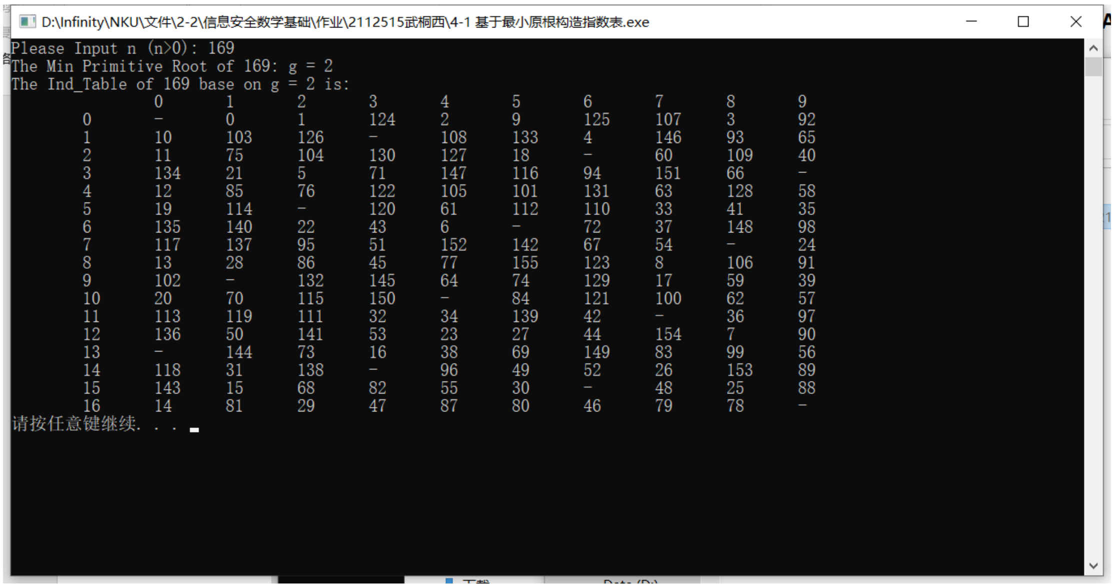
① 可以利用二维数组$A[i][j]$来表示指数表第$i$行第$j$列的值。

② 可以利用平方-乘算法，求解模幂运算$g^r\ (mod\ m)$。

③ 输出指数表时，注意输出的格式。

➢ **运行示例：**



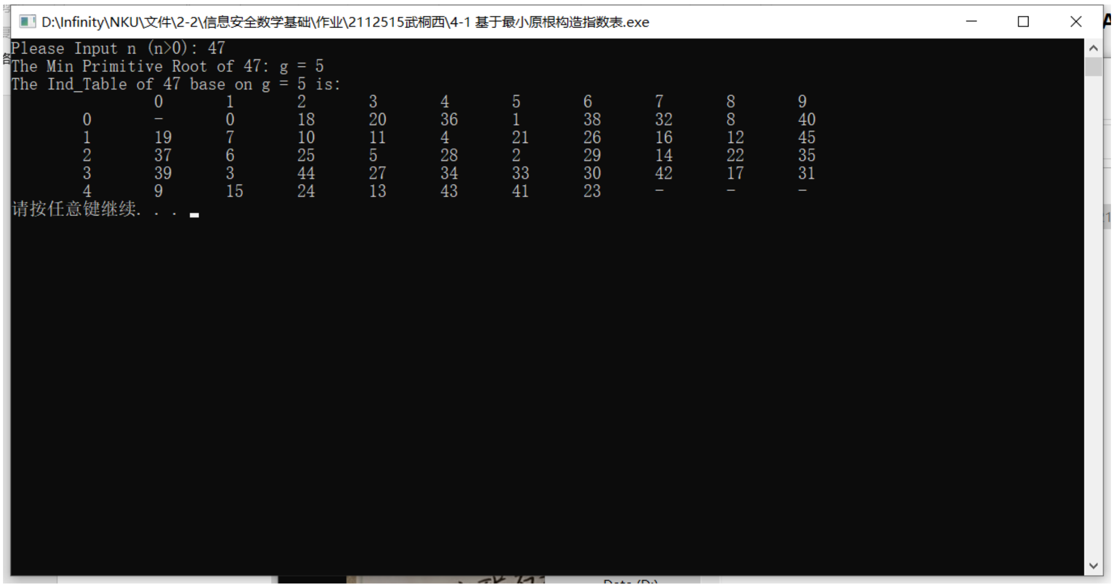


D:\Infinity\NKU\文件\2-2\信息安全数学基础\作业\2112515武桐西\4-1 基于最小原根构造指数表.exe

Please Input n (n>0): 169
The Min Primitive Root of 169: g = 2
The Ind_Table of 169 base 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 | - |

请按任意键继续. . .

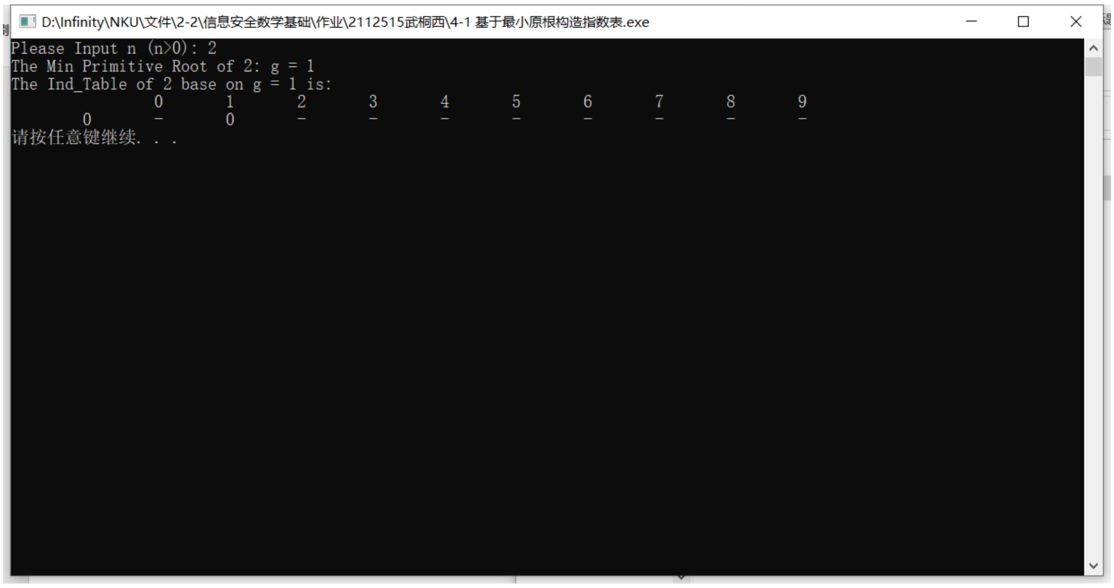


D:\Infinity\NKU\文件\2-2\信息安全数学基础\作业\2112515武桐西\4-1 基于最小原根构造指数表.exe

Please Input n (n>0): 47
The Min Primitive Root of 47: g = 5
The Ind_Table of 47 base on g = 5 is:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | - | 0 | 18 | 20 | 36 | 1 | 38 | 32 | 8 | 40 |
| 1 | 19 | 7 | 10 | 11 | 4 | 21 | 26 | 16 | 12 | 45 |
| 2 | 37 | 6 | 25 | 5 | 28 | 2 | 29 | 14 | 22 | 35 |
| 3 | 39 | 3 | 44 | 27 | 34 | 33 | 30 | 42 | 17 | 31 |
| 4 | 9 | 15 | 24 | 13 | 43 | 41 | 23 | - | - | - |

请按任意键继续. . .

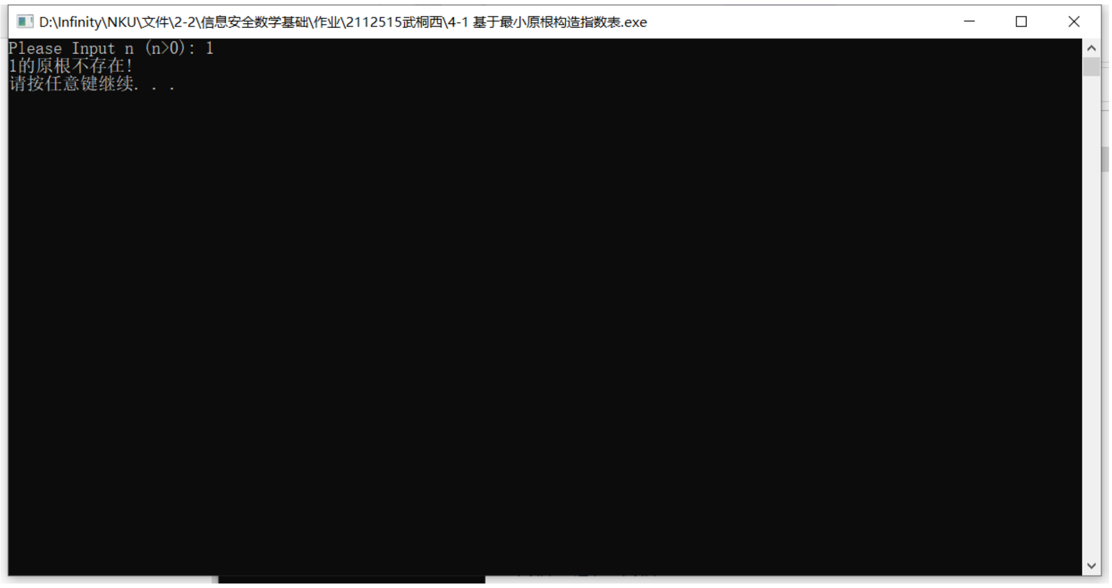


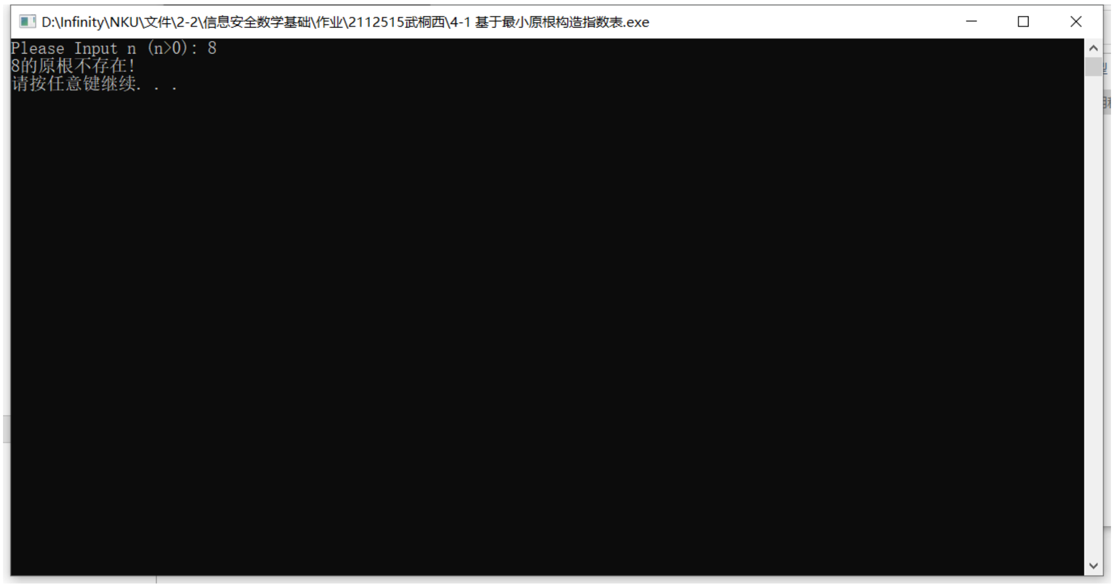
D:\Infinity\NKU\文件\2-2\信息安全数学基础\作业\2112515武桐西\4-1 基于最小原根构造指数表.exe

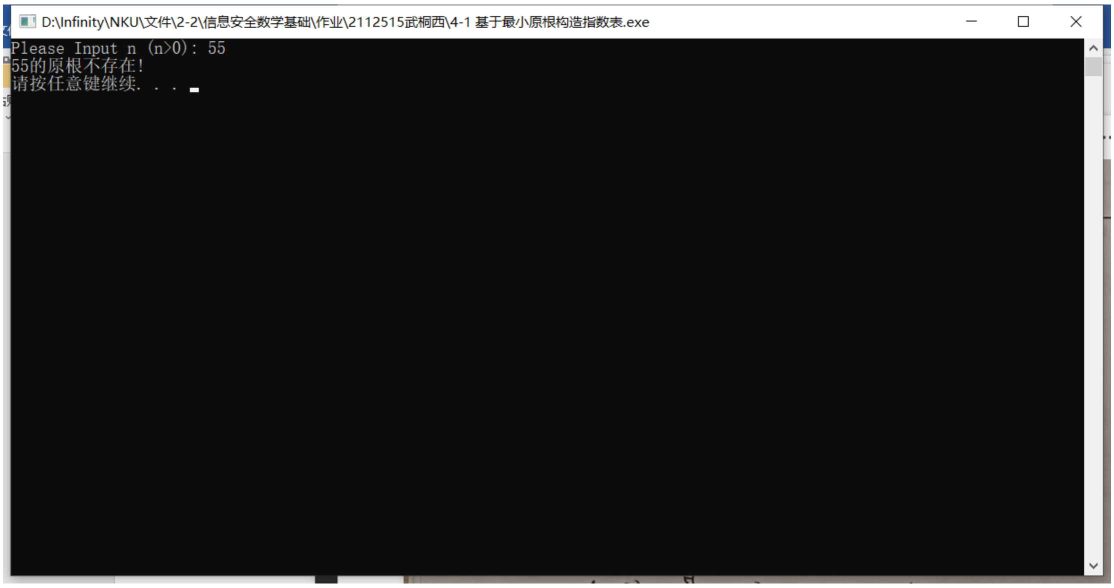
Please Input n (n>0): 2
The Min Primitive Root of 2: g = 1
The Ind_Table of 2 base on g = 1 is:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | - | 0 | - | - | - | - | - | - | - | - |

请按任意键继续. . .


D:\Infinity\NKU\文件\2-2\信息安全数学基础\作业\2112515武桐西\4-1 基于最小原根构造指数表.exe
Please Input n (n>0): 1
1的原根不存在!
请按任意键继续. . .


D:\Infinity\NKU\文件\2-2\信息安全数学基础\作业\2112515武桐西\4-1 基于最小原根构造指数表.exe
Please Input n (n>0): 8
8的原根不存在!
请按任意键继续. . .


D:\Infinity\NKU\文件\2-2\信息安全数学基础\作业\2112515武桐西\4-1 基于最小原根构造指数表.exe
Please Input n (n>0): 55
55的原根不存在!
请按任意键继续. . .