# 第2章 同余 参考答案

## 计算证明

1.计算欧拉函数 $\varphi(n)$：（1）$n=24$ (2) $n=360$

**解** （1）$\varphi(24)=24\times\left(1-\frac{1}{2}\right)\times\left(1-\frac{1}{3}\right)=8$

（2）$\varphi(360)=360\times\left(1-\frac{1}{2}\right)\times\left(1-\frac{1}{3}\right)\times\left(1-\frac{1}{5}\right)=96$

2. 计算：（1）$7^{2023}(mod\,9)$ （2）$666^{666}(mod\,21)$

**解** （1）易知$(7,9)=1$且 $\varphi(9)=6$，则由欧拉定理可知 $7^6\equiv 1(mod\,9)$，故 $7^{2023}=7^{337\times 6+1}\equiv 7(mod\,9)$ .

（2）（不能用欧拉定理和费马小定理）$666^{666}\equiv 15^{666}(mod\,21)$，而由 $15^2\equiv 15(mod\,21)$得$\forall k>1(k\in\mathbb{Z})$，$15^k\equiv 15(mod\,21)$，故 $666^{666}=15(mod\,21)$ .

3. 求解：（1）$x^{86}\equiv 6(mod\,29)$ （2）$x^{21}\equiv 6(mod\,7)$

**解** （1）由费马小定理可知 $x^{29}\equiv x(mod\,29)$，则 $x^2\equiv 6(mod\,29)$，而 $(\pm 8)^2=64=6+29\times 2$，故 $x\equiv\pm 8\equiv 8,21(mod\,29)$ .

（2）由费马小定理可知 $x^7\equiv x(mod\,7)$，则 $x^3\equiv 6(mod\,7)$，遍历完全剩余系得到，$x\equiv 3,5,6(mod\,7)$ .

4. 求模11的一个完全剩余系 $\{r_1,r_2,\cdots,r_i,\cdots,r_{11}\}$ 满足 $\forall i,\ r_i\equiv 1(mod\,3)$ .

**解** 模11的最小非负完全剩余系为 $\mathbb{Z}_{11}=\{0,1,2,\cdots,10\}$ . 由于 $(3,11)=1$，那么模3为1的剩余类由以该剩余系的不同代表元生成，即 $r_i=3(i-1)+1$，得到 $\{1,4,7,10,13,16,19,22,25,28,31\}$ . （答案不唯一）

5. 求 $\sum\limits_{i=1}^{2023} i^{2021}(mod\,4)$ .

**解** 设 $k\in\mathbb{Z}^+$ , 当 $i=2k$ 时，必有 $4\mid i^{2021}$，即 $i^{2021}\equiv 0(mod\,4)$ .

当 $i=2k-1$ 时，有 $(i,4)=1$且$\varphi(4)=2$，由欧拉定理得，$i^2\equiv 1(mod\,4)$，则 $i^{2021}\equiv i(mod\,4)$ .

故 $\sum\limits_{i=1}^{2023} i^{2021}(mod\,4)\equiv\sum\limits_{j=1}^{1012}(2j-1)=1012^2\equiv 0(mod\,4)$ .

6.求105，121的最大公因子、最小公倍数以及相互模的逆元. （无过程不给分）

**解** 由扩展的欧几里得算法进行计算：

| $i$ | $r_i$ | $q_i$ | $s_i$ | $t_i$ |
|---|---|---|---|---|
| **0** | 121 | - | 1 | 0 |
| **1** | 105 | 1 | 0 | 1 |
| **2** | 16 | 6 | 1 | -1 |
| **3** | 9 | 1 | -6 | 7 |
| **4** | 7 | 1 | 7 | -8 |
| **5** | 2 | 3 | -13 | 15 |
| **6** | **1** | 1 | **46** | **-53** |
| **7** | 0 | △ | △ | △ |

$(105,121)=1$

$[105,121]=105\times 121=12705$

$105^{-1}\equiv -53\equiv 68(mod\,121)$

$121^{-1}\equiv 46(mod\,105)$

7. 在某个密码系统中采用参数为 (7,3) 的仿射变换进行加密，即对于明文$x$被加密成密文$y$，满足$y=7x+3(mod\,26)$ . 已知该系统只采用26个小写拉丁字母传递消息，加密时对字母串的每一个字母进行上述仿射变换加密，且有如下对应关系：$a\leftrightarrow 0, b\leftrightarrow 1,\cdots,z\leftrightarrow 25$ 。如对消息 "ac" 加密，$'a'(x=0)\overset{7\times 0+3\equiv 3(mod\,26)}{\Longrightarrow}'d'(y=3)$，$'c'(x=2)\overset{7\times 2+3\equiv 17(mod\,26)}{\Longrightarrow}'r'(y=17)$，密文为"dr". 现在截获到该密码系统传递的密文为"hcxufqvn"，请解密.

**解** 将加密原理变化得到解密操作：$x\equiv 7^{-1}(y-3)\equiv 15(y-3)\equiv 15y+7(mod\,26)$ .

| | |
|---|---|
| $h\leftrightarrow 7,\quad 7\times 15+7\equiv 8\leftrightarrow i$ | $c\leftrightarrow 2,\quad 2\times 15+7\equiv 11\leftrightarrow l$ |
| $x\leftrightarrow 23,\quad 23\times 15+7\equiv 14\leftrightarrow o$ | $u\leftrightarrow 20,\quad 20\times 15+7\equiv 21\leftrightarrow v$ |
| $f\leftrightarrow 5,\quad 5\times 15+7\equiv 4\leftrightarrow e$ | $q\leftrightarrow 16,\quad 16\times 15+7\equiv 13\leftrightarrow n$ |
| $v\leftrightarrow 21,\quad 21\times 15+7\equiv 10\leftrightarrow k$ | $n\leftrightarrow 13,\quad 13\times 15+7\equiv 20\leftrightarrow u$ |

解密得到明文"ilovenku".

8. 求证对 $n\in\mathbb{Z}$，有 $42\mid(n^7-n)$ .

**证明** 由费马小定理有$\begin{cases}n^2\equiv n(mod\,2)\\n^3\equiv n(mod\,3)\\n^7\equiv n(mod\,7)\end{cases}\Rightarrow\begin{cases}n^7\equiv n(mod\,2)\\n^7\equiv n(mod\,3)\\n^7\equiv n(mod\,7)\end{cases}$，得 $n^7\equiv n(mod\,42)$ 即 $42\mid(n^7-n)$ . 证毕.

9. 证明若$p$为素数，且$0<k<p$，则有$(p-k)!\cdot(k-1)!\equiv(-1)^k(mod\,p)$ .

**证明**

$(k-1)! \equiv (-p+k-1)(-p+k-2)\cdots(-p+1) \equiv (-1)^{k-1}[p-(k-1)][p-(k-2)]\cdots(p-1) \pmod p$

$(p-k)!\cdot(k-1)! \equiv (-1)^{k-1}(p-k)![p-(k-1)][p-(k-2)]\cdots(p-1) \equiv (-1)^{k-1}(p-1)! \pmod p$

由威尔逊定理得$(p-1)! \equiv -1 \pmod p$，则$(p-k)!\cdot(k-1)! \equiv (-1)^k \pmod p$.

10. 若$p$为素数，$n$为整数，证明：$p\nmid n$当且仅当 $\varphi(pn)=(p-1)\varphi(n)$ .

**证明**　设 $n=q_1^{\alpha_1}q_2^{\alpha_2}\cdots q_s^{\alpha_s}$，其中 $q_i$ 是素数且 $\alpha_i\neq 0$，$1\le i\le s$ . 易知 $\varphi(n)=n\prod\limits_{i=1}^{s}\dfrac{q_i-1}{q_i}$ .

充分性. 当 $p\mid n$ 时，$\exists\, j,\ p=q_j$，则 $pn=q_1^{\alpha_1}\cdots q_{j-1}^{\alpha_{j-1}}q_{j+1}^{\alpha_{j+1}}\cdots q_s^{\alpha_s}\cdot q_j^{\alpha_j+1}$ ，$\varphi(pn)=p\varphi(n)$. 当 $p\nmid n$时，$\forall\, i,\ p\neq q_i$，则$pn=q_1^{\alpha_1}q_2^{\alpha_2}\cdots q_s^{\alpha_s}\cdot p$ ，$\varphi(pn)=(p-1)\varphi(n)$ . 只有当 $p\nmid n$时成立.

必要性. 由$p\nmid n$得到，$pn=q_1^{\alpha_1}q_2^{\alpha_2}\cdots q_s^{\alpha_s}\cdot p$ ，$\varphi(pn)=(p-1)\varphi(n)$ .

综上，证毕.

11. *（选做）证明正整数 $n$ 和 $n+2$ 是一对孪生素数当且仅当 $4((n-1)!+1)+n\equiv 0(mod\, n(n+2)),\ n\neq 1$ .

**证明**

**必要性.** 由威尔逊定理得，$\begin{cases}(n-1)!\equiv -1 \pmod n & \cdots\cdots(1)\\ (n+1)!\equiv -1 \pmod{n+2} & \cdots\cdots(2)\end{cases}$ . 由(1)易知，$4((n-1)!+1)+n\equiv 0 \pmod n$ .

只需证 $4((n-1)!+1)+n\equiv 0 \pmod{n+2}$ ，而 $(n,n+2)=(n+1,n+2)=1$ ，需证 $4(n+1)!+n(n+1)(n+4)\equiv 0 \pmod{n+2}$ . 而由 (2) 可化简左式得到，$4(n+1)!+n(n+1)(n+4)\equiv 4(n+1)+n(n+1)(n+4)\equiv (n+1)(n+2)^2\equiv 0 \pmod{n+2}$ ，显然成立.

**充分性.** $4((n-1)!+1)+n\equiv 0(mod\, n(n+2)) \Rightarrow \begin{cases}4((n-1)!+1)+n\equiv 0 \pmod n & \cdots\cdots(3)\\ 4((n-1)!+1)+n\equiv 0 \pmod{n+2} & \cdots\cdots(4)\end{cases}$ .

---

为了进一步证明，首先证明**引理：正整数$n(n>1)$满足 $n\nmid(n-1)!$当且仅当$n$为4或素数.**

证明：对$n$所有可能取值的情况进行讨论，分类为：素数、4、其他合数：

1）$n$为素数：易知结论成立.

2）$n$为4：代入得到$4\nmid 5$，结论成立.

3）$n$为其他合数：$\exists\, a,b>1,\ n=ab$ .

- 当 $a=b$ 时，若$a,b$均为奇素数$p$，即 $n=p^2$，则$p^2>2p>p$ ，得$p\cdot 2p\mid(n-1)!$，故$n=p^2\mid(n-1)!$，结论不成立. 若$a,b$合数，可以取$a$为该合数的一个非平凡因子，使$a\neq b$.
- 当 $a\neq b$ 时，一定有$1<a<n,\ 1<b<n$，故$n=ab\mid(n-1)!$，结论不成立.

综上，引理证毕.

---

当$n$取2或4时，代入原式中不成立，故$n\neq 2$且$n\neq 4$.

假设$n$不是素数且$n\neq 4$ ，由引理可知 $n\mid(n-1)!$，则由(3)得到$4\equiv 0 \pmod n \Rightarrow n=2,4$ 矛盾，假设不成立，$n$必为素数.

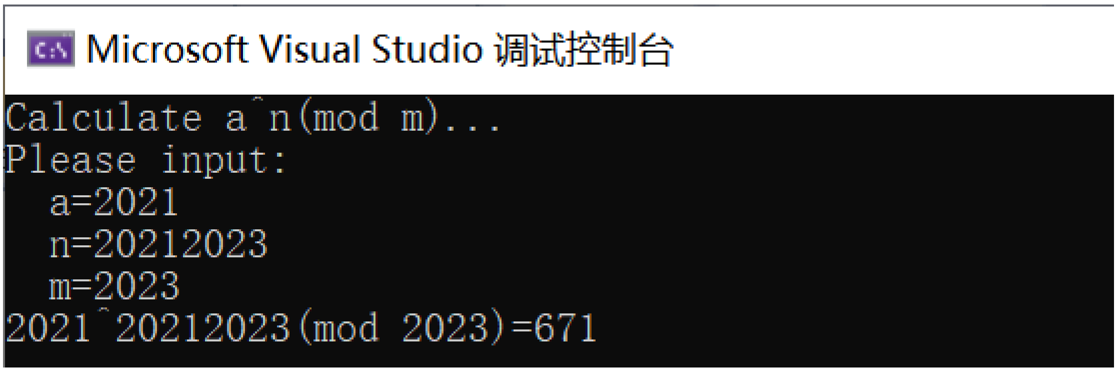
假设 $n+2$ 不是素数且 $n\neq 2$ ，由引理可知 $n+2\mid(n+1)!$ . (4) 左右同乘 $n(n+1)$ 得，$4(n+1)!+n(n+1)(n+4)\equiv 0 \pmod{n+2}$.

而$4(n+1)!+n(n+1)(n+4)\equiv 2n(n+1)\equiv 4(mod\,n+2)\ \Rightarrow\ n=2$ 矛盾，假设不成立，$n+2$必为素数.

综上，证毕.

# 编程练习（基于C/C++）

1. 编程实现平方-乘算法，效果如图所示.



```
Microsoft Visual Studio 调试控制台
Calculate a^n(mod m)...
Please input:
  a=2021
  n=20212023
  m=2023
2021^20212023(mod 2023)=671
```

```cpp
#include<iostream>
using namespace std;

int pow_mod(int a, int n, int m)
{
    int rst = 1;
    while (n > 0)
    {
        if (n & 1)
        {
            rst *= a;
            rst %= m;
        }
        a *= a;
        a %= m;
        n >>= 1;
    }
    return rst;
}

int main()
{
    cout << "Calculate a^n(mod m)..." << endl;
    cout << "Please input:" << endl;
    int a, n, m;
    cout << "  a="; cin >> a;
    cout << "  n="; cin >> n;
    cout << "  m="; cin >> m;
    cout << a << "^" << n << "(mod " << m << ")=" << pow_mod(a, n, m) << endl;
    return 0;
}
```

2. 编程实现扩展的欧几里得算法求逆元，效果如图所示.

Microsoft Visual Studio 调试控制台

```
a=12345
b=65432
gcd(a,b)=1
lcm(a,b)=807758040
a^(-1)=63561(mod 65432)
b^(-1)=353(mod 12345)
```

```cpp
#include<iostream>
using namespace std;

void swap(int& a, int& b)
{
    a = a ^ b;
    b = a ^ b;
    a = a ^ b;
}

int extend_Euclid(int a, int b,int&inv_a,int&inv_b)
{
    if (a < b)return extend_Euclid(b, a, inv_b, inv_a);
    int a0 = a, b0 = b, q = 1;
    int s0 = 1, s1 = 0, t0 = 0, t1 = 1;
    while (a % b != 0)
    {
        q = a / b;
        a = a % b;
        swap(a, b);
        s0 -= q * s1;
        swap(s0, s1);
        t0 -= q * t1;
        swap(t0, t1);
    }
    inv_a = s1 > 0 ? s1 : s1 + b0;
    inv_b = t1 > 0 ? t1 : t1 + a0;
    return b;
}

int main()
{
    int a, b, inv_a, inv_b;
    cout << "a=";
    cin >> a;
    cout << "b=";
    cin >> b;
    int gcd = extend_Euclid(a, b, inv_a, inv_b);
    int lcm = a * b / gcd;
    cout << "gcd(a,b)=" << gcd << endl;
    cout << "lcm(a,b)=" << lcm << endl;
    cout << "a^(-1)=" << inv_a << "(mod " << b << ")" << endl;
    cout << "b^(-1)=" << inv_b << "(mod " << a << ")" << endl;
}
```