Homework 1

Wenye Xiong 2023533141

March 18, 2024

1 Problem 1

```python
def SquareAndMutiply(a, e, n):
    b = bin(e)[2:]
    b = b[::-1]
    ans = 1
    for i in range(len(b)):
        if b[i] == '1':
            ans = (a * ans) % n
        a = (a * a) % n
    return ans

base, exponent, modulus = map(int, input().split())
print(SquareAndMutiply(base, exponent, modulus))
```

The result is 19489389945386041607071081817241920919542635233623116738469155055206259159226436938865465087133511096927509156841578783141212143489199923529097996539792654733505278706812520830942209991900318336435802408907249020763770922682237250909513951994814724102553142432605916650209186930443817371994324442380618239060899770209698997113410596399791595727394196009053367816731883686504687107181648321094994097671995305419040805120814031555590587098823477471474182303588141313811472082913287478579910489774659842657219793245954171847503170017151440737380478840189460378458005476484742953848813170374548455806977675820760128018344

2 Problem 2

```python
def extended_euclidean_algorithm(a, b):
    if b == 0:
        return 1, 0
    else:
        s, t = extended_euclidean_algorithm(b, a % b)
        return t, s - (a // b) * t

a, b = map(int, input().split())
s, t = extended_euclidean_algorithm(a, b)
print(s, t)
```

The result is:
s = 526934651740475975791740640830612065757613986569351144308112435606950663069562377006384677413803445132609836259065451941548001267078692425281992503034711715362075978960084056501348894581563254902960363363426447969584774252883983875181782658907006563057148373685234965973219732121971442442376472912705292015898
t = -4922435602557020575264036911319758978419249536244008420108775719343721274111896002459291667895080234292453411578954324261793651077186663625890948400350842512853060168116459859792483937224361285850400246381718448690438802997126844191121984884459076214105581336516953336118974124756550236257925745365828061 3873

3 Problem 3

3.1 1

Consider the induction on i:

For i = 0, the statement is clear. $s_0 = 1, t_0 = 0, s_1 = 0, t_1 = 1$, $s_0 t_1 - t_0 s_1 = 1$

For i = 1, 2,...., k, we have:

$$\begin{aligned} s_i t_{i+1} - t_i s_{i+1} &= s_i(t_{i-1} - t_i q_i) - t_i(s_{i-1} - s_i q_i) \\ &= s_i t_{i-1} - t_i s_{i-1} \\ &= -(s_{i-1} t_i - t_{i-1} s_i) \end{aligned}$$

So according to the induction,

$$\begin{aligned} s_i t_{i+1} - t_i s_{i+1} &= -(-1)^{i-1} \\ &= (-1)^i \end{aligned}$$

3.2 2

We can also easily prove both statements by induction on i:

Both statements are obviously true for i = 0: $t_0 = 0, t_1 = 1, t_0 t_1 = 0, |t_0| \leq |t_1|$

For i = 1,....,k, we have $t_{i+1} = ti - 1 - t_i q_i$
And by the induction hypothesis, t_{i-1}, t_i have opposite signs and $|t_i| \geq |t_{i-1}|$

So it leads to $|t_{i+1}| = |t_{i-1}| + |t_i| q_i$
Because $q_i \geq i$, so $|t_{i+1}| \geq |t_i|$. Plus that $t_{i+1} = ti - 1 - t_i q_i$, t_{i-1}, t_i have opposite signs, $|t_{i+1}|$ and $|t_i|$ also have opposite signs, which leads to $t_{i+1} t_i \leq 0$.

4 Problem 4

First we consider the two equations:

$$as_{i-1} + bt_{i-1} = r_{i-1}$$
$$as_i + bt_i = r_i$$

Subtracting t_{i-1} times the second equation from t_i times the first, we get:

$$as_{i-1}t_i - as_it_{i-1} = r_{i-1}t_i - r_it_{i-1}$$

According to the result of problem 3, we have $s_{i-1}t_i - t_{i-1}s_i = (-1)^i$, apply this equation, we get:

$$r_{i-1}t_i - r_it_{i-1} = \pm a$$

Using the result of problem 3, t_i and t_{i-1} have opposite signs, we have:

$$a = r_{i-1}|t_i| + r_i|t_{i-1}|$$

Obviously, $a \geq r_{i-1}|t_i|$ for i = 1,2,....k+1.

Follow from this, because $a > 0$, then $r_{i-1} > 0$ for i=1,2,....k+1.
r_{i-1} is an integer, so $r_{i-1} \geq 1$ for i=1,2,....k+1. That means $|t_i| \leq a$ for i=1,2,....k+1.

5 Problem 5

To determine the set of Fermat liars for n=21, we first consider the factors of 21, which are 3 and 7.

If $a^{20} \equiv 1 \pmod{21}$, then $a^{20} \equiv 1 \pmod 3$ and $a^{20} \equiv 1 \pmod 7$.
According to Fermat's little theorem, $a^2 \equiv 1 \pmod 3$ and $a^6 \equiv 1 \pmod 7$ for all integers $a \in [1, n-1]$ and cannot divide 3 or 7

So to satisfy the equation $a^{20} \equiv 1 \pmod 3$, a must be in $[1]_3$ or $[2]_3$. To satisfy the equation $a^{20} \equiv 1 \pmod 7$, $a^2 \equiv 1 \pmod 7$. Which means a must be in $[1]_7$ or $[6]_7$. These include 1,6,8,13,15,20.

For 1,6,8,13,15,20, examine if they are in $[1]_3$ or $[2]_3$. Finally we get 1,8,13,20 which are Fermat Liars.

Then according to Chinese Reminder Theorem, the answer is 1,8,13,20.

6 Problem 6

We observe that $ax \equiv b \pmod n \iff n|ax-b \iff (\frac{n}{d})|[(\frac{a}{d})x - (\frac{b}{d})]$
That is, x is a solution of $ax \equiv b \pmod n$ if and only if x is a solution of $(\frac{a}{d})x \equiv (\frac{b}{d}) \pmod{(\frac{n}{d})}$

Now, because $d = gcd(a, n)$, $\frac{a}{d}$ and $\frac{n}{d}$ are relatively prime. So there is only one residue class $t = (\frac{a}{d})^{-1} \pmod{\frac{n}{d}}$

So s= $\frac{b}{d}t$ is a solution of $(\frac{a}{d})x \equiv (\frac{b}{d}) \pmod{(\frac{n}{d})}$, and also a solution of $ax \equiv b \pmod n$
Consider the residue classes $s, s+n/d, s+2n/d, ..., s+(d-1)n/d$, they are all solutions of $ax \equiv b \pmod n$.So last thing is we need to prove that $s+(d-1)n/d < n$
That is to prove $s < \frac{n}{d}$

Suppose that s is the smallest number to satisfy $(\frac{a}{d})x \equiv (\frac{b}{d}) \pmod{(\frac{n}{d})}$, if $s \geq \frac{n}{d}$,then we have:

$$(s - \frac{n}{d})\frac{a}{d} = \frac{a}{d}s - \frac{a}{d}\frac{n}{d} \equiv (\frac{b}{d}) \pmod{(\frac{n}{d})}$$

So s must be smaller than $\frac{n}{d}$, and the proof is complete.

After all, among all the residue classes modulo n, the residue classes represented by

$$\frac{b}{d}t, \frac{b}{d}t + \frac{n}{d}, \frac{b}{d}t + 2\frac{n}{d}, ..., \frac{b}{d}t + (d-1)\frac{n}{d}$$

are the only ones that are solutions of $ax \equiv b \pmod n$.