

Solution 7.27

T1 Offer

按题意模拟即可， 签到题

【核心代码】

```
while(n--) {
    memset(cnt, 0, sizeof cnt);
    string str;
    cin >> str;
    for(int i = 0; i < str.size(); i++)
        cnt[str[i]-'A']++;
    if(cnt[3] > 0 || cnt[2] > 1)
        cout << "failed" << endl;
    else if(cnt[0] > 2)
        cout << "sp offer" << endl;
    else
        cout << "offer" << endl;
}
```

T2: 简单筛法

注意到虽然要找的素数的范围很大，但是最大的区间只有 1000000

因此我们可以对区间进行素数筛法，筛的时候注意区间平移，比如 [5, 10] 我们可以平移为[1, 6] 也就是这里的 1 代表了 5， 6 代表了 10.

于是我们可以进行 T 次 素数筛法

【核心代码】

```
void segment_sieve(LL a, LL b) {
    for (int i = 0; (LL)i * i < b; i++) is_prime_small[i] = true;
    for (int i = 0; i < b - a; i++) is_prime[i] = true;
    for (int i = 2; (LL)i * i < b; i++) {
        if (is_prime_small[i]) {
            for (int j = 2 * i; (LL)j * j < b; j += i) {
                is_prime_small[j] = false; //筛[2,sqrt(b))
            }
            for (LL j = max(2LL, (a + i - 1) / i) * i; j < b;
                 j += i) //((a+i-1)/i 为[a,b)区间内的第一个数至少为 i 的多少倍.
            {
                is_prime[j - a] = false; //筛[a,b)
            }
        }
    }
}
```

T3: 薪资

标签：贪心

由贪心可知，绩效得 C 的人一定是初始薪资最少的人。其他人的薪资都 *2 或 *3，他即使被开除也无所谓。

其他人的薪资用快速幂算出即可。

【核心代码】

```
sort(co+1,co+1+n);
reverse(co+1, co+1+n);
long long ans = 0;
for (int i = 1; i <= x; ++i)
    ans = (ans + co[i] * pow_mod(3, m)) % mod;

For (int i = x + 1; i <= x + y; ++i)
    ans = (ans + co[i] * pow_mod(2, m)) % mod;

if(m < 2)
    for (int i = x + y + 1; i <= n; ++i)
        ans += co[i];
cout << ans << endl;
```

T4 变换

20pt: 两个数字一定是变为它们的 gcd，求一下它们到 gcd 需要几次操作即可

40pt: 如果可以发现“将所有数字变小直至相等”的效果等同于“将数字变大直至相等”，那么可以枚举最终的数字

另外 20pt: 值域很小时，最终变成的数字 Q 的范围也会很小，我们可以枚举 Q 的范围 + 剪枝，然后看看其他数字变成 Q 需要几次操作。

100pt:

我们可以只关注操作 1，将一个数字不变，其他数字乘 x，等价于其他数字不变，这个数字除以 x，其中 x 为这个数字的一个素因子。

题目等价于每次可以将一个数字除以 x (x 为其素因子)，问多少次可以将所有数字全部变成一样的。

易得，我们将所有数字都变成他们的 gcd 就行了，我们求出所有数字的 gcd，让所有数字都除以 gcd，然后再算出所有数字的素因子个数和即可。

可以对上述操作进行一些优化或者预处理，使之能够通过全部数据。

对于质因子 j ，所有数里面质因子 j 出现的总和是 $\text{cnt}[j]$ ，在一个元素里最少出现次数为 $\text{min}[j]$

1. 如果每个元素都有质因子 j ，那么我们的操作总数为 $\text{cnt}[j] - n * \text{min}[j]$
2. 如果存在至少一个元素里面不存在质因子 j ，那么我们的操作总数就是 $\text{cnt}[j]$