```python
def modular_inverse(a: int, m: int) -> int:
    m0, t1, t2 = m, 0, 1
    if m == 1:
        return 0
    while a > 1:
        q = a // m
        m, a = a % m, m
        t1, t2 = t2 - q * t1, t1
    if t2 < 0:
        t2 += m0
    return t2

def chinese_remainder(modulus, remainders):
    N = 1
    for mod in modulus:
        N *= mod

    x = 0
    for mod, rem in zip(modulus, remainders):
        Ni = N // mod
        inverse = modular_inverse(Ni, mod)
        x += rem * Ni * inverse

    return x % N

modulus = [3, 5, 7]
remainders = [2, 3, 2]

result = chinese_remainder(modulus, remainders)
print("The solution is:", result)


OUTPUT:
The solution is: 23
```