

Part 1:

```
> restart;
> unprotect(CRT):
CRT := proc(u_List::list, m_List::list, var_Indet::name, p::posint)
    local n, rec_Call, a_One, m_One, a_Two, m_Two, mod_Inverse, s,
    t, cong_Rem, prod_Mod;

    n := nops(u_List);
    if nops(m_List) <> n then
        return NULL:
    end if;

    if n = 0 then
        return NULL:
    elif n = 1 then
        return [u_List[1], m_List[1]];
    else
        rec_Call := CRT(u_List[1 .. -2], m_List[1 .. -2], var, p);

        a_One := rec_Call[1];
        m_One := rec_Call[2];
        a_Two := u_List[-1];
        m_Two := m_List[-1];

        mod_Inverse := Gcdex(m_One, m_Two, var_Indet, 's', 't') mod
p;

        if mod_Inverse <> 1 then
            return NULL:
        end if:

        cong_Rem := Rem(expand(a_One + (a_Two - a_One)*s*m_One ),
m_One*m_Two, var_Indet) mod p;
        prod_Mod := expand(m_One * m_Two) mod p:

        return [cong_Rem, prod_Mod];
    end if;
end proc:

> u := [y^2, y^2 + y + 1, y^3]:
m := [y^3 + y + 1, y^3 + y^2 + 1, y^4 + y + 1]:

CRT(u, m, y, 2);
[y^9 + y^6 + y^5 + y^4 + y^3 + y + 1, y^10 + y^9 + y^8 + y^6 + y^5 + y^4 + 1]
> (* Testing this procedure for Q4 on A2 *)
```

(1)

u_New := [y^2, y^2 + y + 1]:

m_New := [y^3 + y + 1, y^3 + y^2 + 1]:

CRT(u_New, m_New, y, 2);

$[y^5 + y^4 + y^3 + y^2 + y, y^6 + y^5 + y^4 + y^3 + y^2 + y + 1]$

(2)