> Math 418 : MAPLE Solution of Assignment #9 -- *Your NAME*

```
> restart; with(numtheory) :
```

Problem 4: Elliptic curve programs

(a) Program to calculate the number of points on the elliptic curve
y^2 = x^3 + ax +b, with a, b \in F_p.

Note: This uses the legendre function from the numtheory package

```
> numE := proc(a, b, p) local x;
    return(1 + p + sum(legendre(x^3 + a·x + b, p), x = 1 ..p)); end;
```

$$numE := \mathbf{proc}(a, b, p) \quad \textbf{(1)}$$

```
    local x;
    return 1 + p + sum(numtheory:-legendre(x^3 + a * x + b, p), x = 1 ..p)
end proc
```

(b) Test the above program for the elliptic curve
E/F_773 : y^2 = x^3 + 5x + 3.

```
> p := 773 : N773 := numE(5, 3, p);
```

$$N773 := 804 \quad \textbf{(2)}$$

Thus, |E(F_773)| = 804, i.e., E has 804 points over F_773.

Check Hasse's inequality:

```
> aE := p + 1 - N773;  evalf(2· sqrt(p));
```

$$aE := -30$$

$$55.60575510 \quad \textbf{(3)}$$

Clearly, |aE| = 30 < 55 < 2sqrt(p). We can also check this by using Maple:

```
> evalb(abs(aE) < evalf(2·sqrt(p)));
```

$$true \quad \textbf{(4)}$$

(c) Program for the addition of points on the elliptic curve
y^2 = x^3 + ax + b mod p

- includes point doubling (for this, need coefficient a)
- use [] to denote the point at infinity

```
> addpts := proc(p1, p2, a, p) local lm, x3, y3;
    if p1 = [ ] then return(p2); fi;
    if p2 = [ ] then return(p1); fi;
    if modp(p1[1] - p2[1], p) = 0
      then if modp(p1[2] + p2[2], p) = 0 then return([ ]);
           else lm := (3·p1[1]^2 + a)/(2·p1[2]) fi;
      else lm := (p2[2] - p1[2])/(p2[1] - p1[1]); fi;
    x3 := lm·lm - p1[1] - p2[1];
    y3 := lm·(p1[1] - x3) - p1[2];
    return([modp(x3, p), modp(y3, p)]); end;
```

$$addpts := \mathbf{proc}(p1, p2, a, p) \quad \textbf{(5)}$$

```
    local lm, x3, y3;
    if p1 = [ ] then return p2 end if;
    if p2 = [ ] then return p1 end if;
```

```
    if modp(p1[1] - p2[1], p) = 0 then
        if modp(p1[2] + p2[2], p) = 0 then
            return [ ]
        else
            lm := 1/2 * (3 * p1[1]^2 + a) / p1[2]
        end if
    else
        lm := (p2[2] - p1[2]) / (p2[1] - p1[1])
    end if;
    x3 := lm * lm - p1[1] - p2[1];
    y3 := lm * (p1[1] - x3) - p1[2];
    return [modp(x3, p), modp(y3, p)]
end proc
```

(d): Testing the above program for the given values:

(i), (ii) Compute the sum of points on E: y^2 = x^3 + 17 over F_31

```
> p1 := [-2, 3]:
```

Check that this point lies on the given curve and that E is an elliptic curve (**not asked: optional**)

```
> chp := (p1, a, b, p) → evalb(modp(p1[1]^3 + a·p1[1] + b - p1[2]^2, p) = 0):
  chE := (a, b, p) → evalb(modp(4·a^3 + 27·b^2, p) ≠ 0):
> chE(0, 17, 13), chp(p1, 0, 17, 31), chp(p2, 0, 17, 31), chp(p3, 0, 17, 31);
                        true, true, true, true                    (6)
> addpts(p1, p1, 0, 31), addpts(p1, [ ], 0, 31);
                        [8, 8], [-2, 3]                           (7)
```

Note: p1 = [-2,3] = [29,3], so the answer [29,3] is also correct.

(iii) Compute the sum of points q_1 + q_i on E: y^2 = x^3 + 50x + 5 over F_97

```
> P1 := [8, 23]: P2 := [2, 4]: a := 50: b := 5: p := 97:
> chE(50, 5, 97), chp(P1, a, b, p), chp(P2, a, b, p);
                        true, true, true                          (8)
> addpts(P1, P2, a, p);
                        [62, 0]                                   (9)
>
```