1. Let E be the elliptic curve y^2 = x^3 + 2x+3(mod 19)
a)
Find the sum (1, 5) + (9, 3)

```
→ b:2;
(%o1) 2
→ c:3;
(%o2) 3
→ n:19;
(%o3) 19
→ x1:1;
(%o4) 1
→ x2:9;
(%o5) 9
→ y1:5;
(%o6) 5
→ y2:mod(3, n);
(%o7) 3
→ d1:mod(y2−y1, n);
(%o8) 17
→ d2:x2−x1;
(%o9) 8
→ gcd(d2, n);
(%o10) 1
→ d2:inv_mod(d2, n);
(%o11) 12
→ m:mod(d1·d2, n);
(%o12) 14
→ x3:mod(mod(m^2, n)−x1−x2, n);
(%o13) 15
→ y3:mod(m·(x1−x3)−y1, n);
(%o14) 8
```

b)
find the sum (9, 3) + (9, -3)

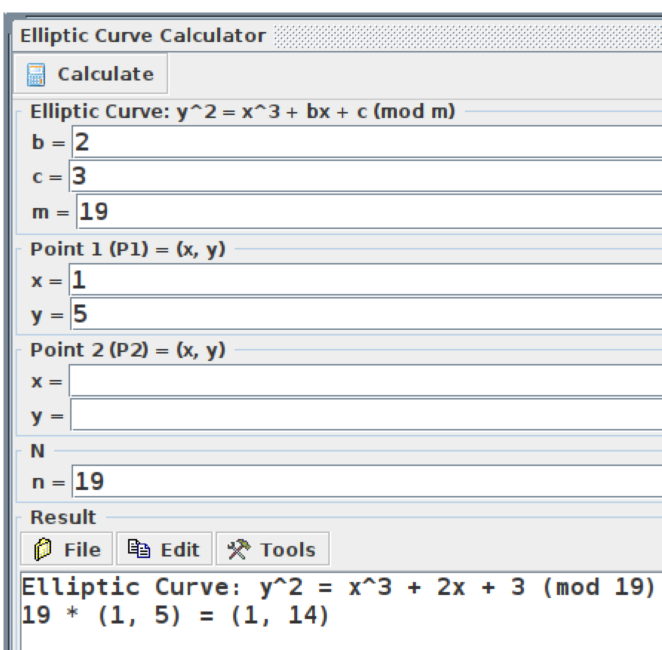
```
→ x1:9;
(%o15) 9
→ x2:9;
(%o16) 9
→ y1:3;
(%o17) 3
→ y2:mod(−3, n);
(%o18) 16
→ d1:mod(y2−y1, n);
(%o19) 13
→ d2:mod(x2−x1, n);
(%o20) 0
→ gcd(d2, n);
(%o21) 19
→ d2:inv_mod(d2, n);
(%o22) false
```

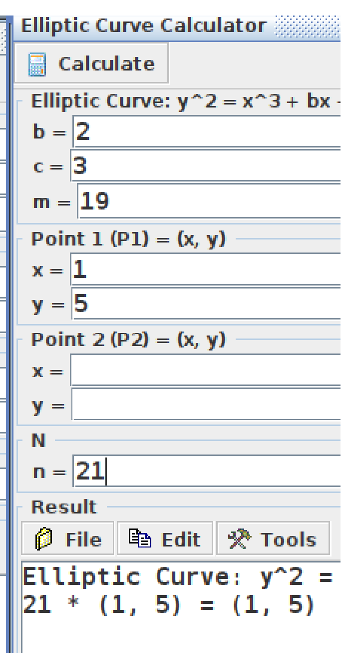
Since the points are directly above each other, The sum is infinity

c)
using the fact that (9, 3) + (9, -3) is infinity, we see that (9, -3) is an additive inverse for (9, 3) so to do the following we simply
Find (1, 5) – (9, 3)

```
         y2:mod(−3, n);
(%o173)  16

(%i174)  d1:mod(y2−y1, n);
(%o174)  11

(%i175)  d2:x2−x1;
(%o175)  8

(%i176)  gcd(d2, n);
(%o176)  1

(%i177)  d2:inv_mod(d2, n);
(%o177)  12

(%i178)  m:mod(d1·d2, n);
(%o178)  18

(%i179)  x3:mod(mod(m^2, n)−x1−x2, n);
(%o179)  10

(%i180)  y3:mod(m·(x1−x3)−y1, n);
(%o180)  4
```

d)
as we can see at 20 we hit infinity and cycle back to (1, 5) so we now have the number of distinct multiples including infinity



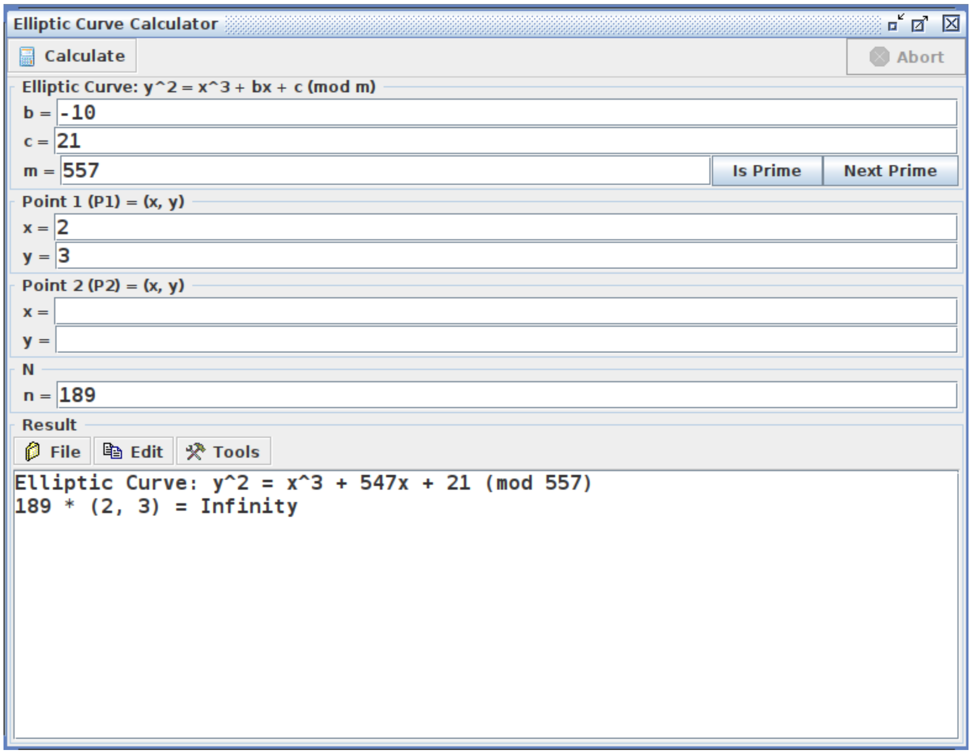


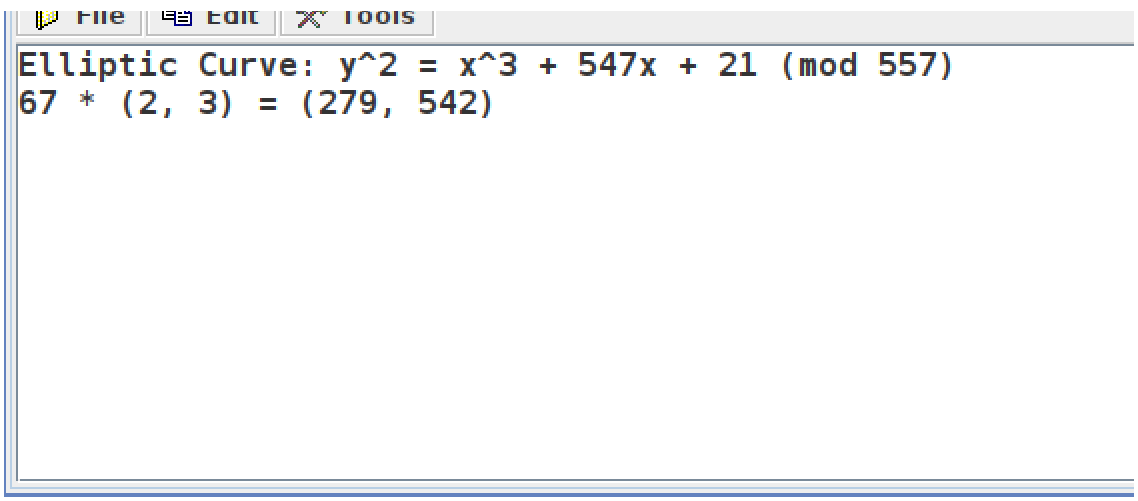
e)

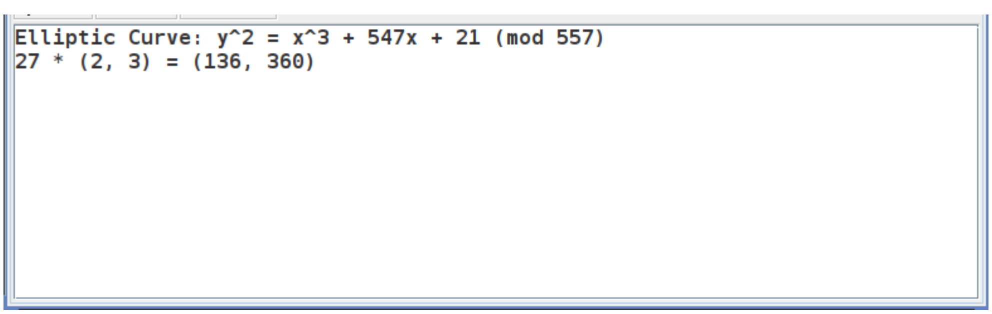
```
(%i333) n+1+2·float(sqrt(n));
(%o333) 28.71779788708134

(%i334) n+1−2·float(sqrt(n));
(%o334) 11.28220211291865
```

11 <= 20 <= 28
20-19-1 < 28

4)
The order of P is189 since it is the smallest number that k*p = infinity
(k/p)P != infinity







```
(%i8)  floor(558+2·float(sqrt(189)));
(%o8)  585

(%i9)  floor(558−2·float(sqrt(189)));
(%o9)  530

(%i18) for i: 530 thru 585 do if mod(i, 189) = 0 then display(mod(i, 189));
       mod(567,189)=0
(%o18) done
```

As you can see, the only number that the order of p divides is 567 so 567 must be the number of points

5)

```
→         p:593899;
(%o291)   593899

→         x1:5;
(%o292)   5

→         x2:1;
(%o293)   1

→         y1:9;
(%o294)   9

→         y2:593898;
(%o295)   593898

→         mn: y2−y1;
(%o296)   593889

→         md:x2−x1;
(%o297)   −4

→         md:inv_mod(593895, p);
(%o298)   445424

→         mod(593895·445424, p);
(%o299)   1

→         m:mod(mn·md, p);
(%o300)   296952

→         x3:mod(m·m, p)−x1−x2;
(%o301)   148475

→         y3:mod(m·(x1−x3)−y1, p);
(%o302)   222715
```

2) represent 12345 as a ciphertext so 12345_ and output was (123453, 65243)

y^2 = x^3 + 7x + 11 (mod 593899)

```
(%i30) m:123453;
(%o30) 123453

(%i31) b:7;
(%o31) 7

(%i32) c:11;
(%o32) 11

(%i33) n:593899;
(%o33) 593899

(%i34) r:mod(m·m·m+b·m+c, n);
(%o34) 174916

(%i35) mod(n, 4);
(%o35) 3

(%i36) y:power_mod(r, (n+1)/4, n);
(%o36) 528656

(%i37) mod(−y, n);
(%o37) 65243
```

3) Below is the output from my factoring program using elliptic curves. Here is also some maxima calculations I started with then the p-1 took much longer to computer than elliptic curve.

```
Please enter a n value to factor:
3900353
3900353 = 1109 * 3517
```

```
→ n:3900353;
(%o1) 3900353

→ b:7;
(%o2) 7

→ x1:2;
(%o3) 2

→ y1:7;
(%o4) 7

→ c:mod(y1^2−x1^3−b, n);
(%o5) 34

→ d1:((3·x1)^2)+b;
(%o6) 43

→ d2:mod(2·y1, n);
(%o7) 14

→ gcd(d2, n);
(%o8) 1

→ d2:inv_mod(d2, n);
(%o9) 835790

→ m:d1·d2;
(%o10) 35938970

→ x2:mod(m^2−x1−x1, n);
(%o11) 616898
```


```
→ x2:mod(m^2−x1−x1, n);
(%o11) 616898
```


```
→ y2:mod(m·(x1−x2)−y1, n);
(%o12) 2005594

→ a:2;
(%o13) 2

→ for i: 1 thru 1000 do (
    if gcd(power_mod(a, i!, n)−1, n) != 0 then (
      display(gcd(power_mod(a, i!, n)−1, n))
    )
  );
(%o27) done
```