```
> ReadFromWeb:=proc(URL::string, {printfile::truefalse:=false})
>  local n,m, status, webfile, headers;
>  status,webfile,headers:=HTTP[Get](URL):
>  if ( HTTP[Code](status) <> "OK") then
>      error(HTTP[Code](status),URL);
>   fi;
>  # now read the web page
>  n:=0:
>  while (n < length(webfile)) do
>    m:=n;
>    parse(webfile,statement,lastread='n', offset=n);
>    if (printfile) then printf("%s",webfile[m+1..n]); fi;
>  od:
> end:
```

```
> prefix:="http://www.math.sunysb.edu/~scott/mat331.
  spr12/problems/":
```

```
> Junk:= [ 2, 4, 7, 9, 15, 22, 11, 31769];
```

$$Junk := [2, 4, 7, 9, 15, 22, 11, 31769] \quad (1)$$

```
> isprime(7);
```

$$true \quad (2)$$

```
> isprime(9);
```

$$false \quad (3)$$

```
> for i from 1 to 10 do
    print (i, i^2);
  od:
```

```
1, 1
2, 4
3, 9
4, 16
5, 25
6, 36
7, 49
8, 64
9, 81
10, 100          (4)
```

```
> for i from 1 to 10 by 2 do
    print (i, i^2);
  od:
```

```
1, 1
3, 9
5, 25
7, 49
9, 81            (5)
```

```
> for i from 1 to nops(Junk) do
    if isprime(Junk[i]) then
       print(Junk[i]);
     fi;
  od;
                                    2
                                    7
                                    11
                                  31769                                  (6)
> JunkyPrimes:=[];
  for i from 1 to nops(Junk) do
    if isprime(Junk[i]) then
      JunkyPrimes:= [ op(JunkyPrimes), Junk[i]];
     fi;
  od;
                             JunkyPrimes := [ ]                          (7)
> JunkyPrimes;
                            [2, 7, 11, 31769]                            (8)
> select(isprime,Junk);
                            [2, 7, 11, 31769]                            (9)
> remove(isprime,Junk);
                              [4, 9, 15, 22]                             (10)
> Junk;
                      [2, 4, 7, 9, 15, 22, 11, 31769]                    (11)
> 14 mod 2;
                                    0                                    (12)
> 15 mod 2;
                                    1                                    (13)
> modp(14,2);
                                    0                                    (14)
> modp(15,2);
                                    1                                    (15)
> select( (modp(x,2)=0), Junk);
                                   [ ]                                   (16)
> select( x->(modp(x,2)=0), Junk);
                                [2, 4, 22]                               (17)
> select( x->(modp(x,2)<>0), Junk);
                          [7, 9, 15, 11, 31769]                          (18)
> isodd:= x-> if(modp(x,2)<>0) then true else false fi;
         isodd := x→if modp(x, 2) ≠ 0 then true else false end if        (19)
> select(isodd, Junk);
                          [7, 9, 15, 11, 31769]                          (20)
> isMaybeEven := x-> modp(x,2);
                       isMaybeEven := x→modp(x, 2)                       (21)
> select(isMaybeEven,Junk);
Error, invalid boolean expression: 0
```

```
> ToyProblem := [ {a=3,b=1,r=0}, {a=8,b=6,r=2.1}, {a=3,b=1,r=0},
  {a=4,b=9, r=-2.1}];
```

$$ToyProblem := [\{a=3, b=1, r=0\}, \{a=8, b=6, r=2.1\}, \{a=3, b=1, r=0\}, \{a=4, b=9, r=-2.1\}] \quad (22)$$

HINT:

```
> subs( {a=3,b=1,r=-2.1}, r);
```

$$-2.1 \quad (23)$$

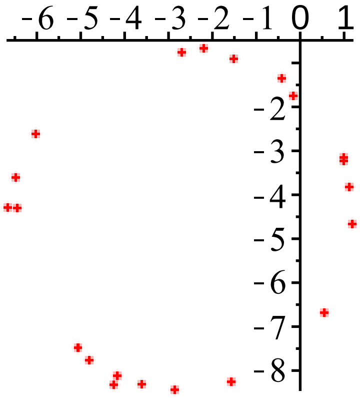
```
> for i from 1 to nops(ToyProblem) do
     if  (subs( ToyProblem[i], r) > 0) then
       ans:=ToyProblem[i];
     fi;
  od;
> ans;
```

$$\{a=8, b=6, r=2.1\} \quad (24)$$

```
> select( x -> (subs( x, r) > 0), ToyProblem);
```

$$[\{a=8, b=6, r=2.1\}] \quad (25)$$

```
> ReadFromWeb(cat(prefix,"lsq_data.txt"));
defined set_seed(s), line_pts(), bad_line_pts(), quadratic_pts
(), exp_pts(), cubic_pts(), and circle_pts()
> cpts:=circle_pts():
  plot(cpts,style=point);
```



```
> fitCirc:= proc(data)
    local epsilon, H, a,b,r,pt,i, ans;
    epsilon:=(a,b,r,pt)-> ( (a-pt[1])^2 + (b-pt[2])^2-r^2)^2;
    H:=(a,b,r,joe) -> sum( epsilon(a,b,r,joe[i]),
                              i=1..nops(joe));
    ans:=solve({diff(H(a,b,r,data),a)=0,
               diff(H(a,b,r,data),b)=0,
               diff(H(a,b,r,data),r)=0});
    select(x->(subs( x, r) > 0),ans);
  end:
```

```
> fitCirc(cpts);
Error, (in fitCirc) selecting function must return true or false
```

```
> fitCirc:= proc(data)
    local epsilon, H, a,b,r,pt,i, ans;
    epsilon:=(a,b,r,pt)-> ( (a-pt[1])^2 + (b-pt[2])^2-r^2)^2;
    H:=(a,b,r,joe) -> sum( epsilon(a,b,r,joe[i]),
                           i=1..nops(joe));
    ans:=solve({diff(H(a,b,r,data),a)=0,
          diff(H(a,b,r,data),b)=0,
          diff(H(a,b,r,data),r)=0});
    # select(x->(subs( x, r) > 0),ans);
  end:
```

```
> evalf(fitCirc(cpts),2);
```

$$\{a=-2.4+3.7\,\mathrm{I},\, b=-4.9+4.5\,\mathrm{I},\, r=0.\},\ \{a=-2.5+3.9\,\mathrm{I},\, b=-4.7-3.3\,\mathrm{I},\, r=0.\},\ \{a=-2.6,\, b=-4.6,\, r=0.\},\ \{a=-2.5-3.9\,\mathrm{I},\, b=-4.7+3.3\,\mathrm{I},\, r=0.\},\ \{a=-2.4-3.7\,\mathrm{I},\, b=-4.9-4.5\,\mathrm{I},\, r=0.\},\ \{a=-2.7,\, b=-4.6,\, r=3.9\},\ \{a=-2.7,\, b=-4.6,\, r=-3.9\} \qquad \text{(26)}$$

```
> fitCirc:= proc(data)
    local epsilon, H,pt,i, ans;
    epsilon:=(a,b,r,pt)-> ( (a-pt[1])^2 + (b-pt[2])^2-r^2)^2;
    H:=(a,b,r,joe) -> sum( epsilon(a,b,r,joe[i]),
                           i=1..nops(joe));
    ans:=solve({diff(H(a,b,r,data),a)=0,
          diff(H(a,b,r,data),b)=0,
          diff(H(a,b,r,data),r)=0});
    select(x->(subs( x, r) > 0), [ans]);
  end:
```

```
> mine:=fitCirc(cpts);
```

$$mine := [\{a=-2.682127600,\, b=-4.568186883,\, r=3.877311723\}] \qquad \text{(27)}$$

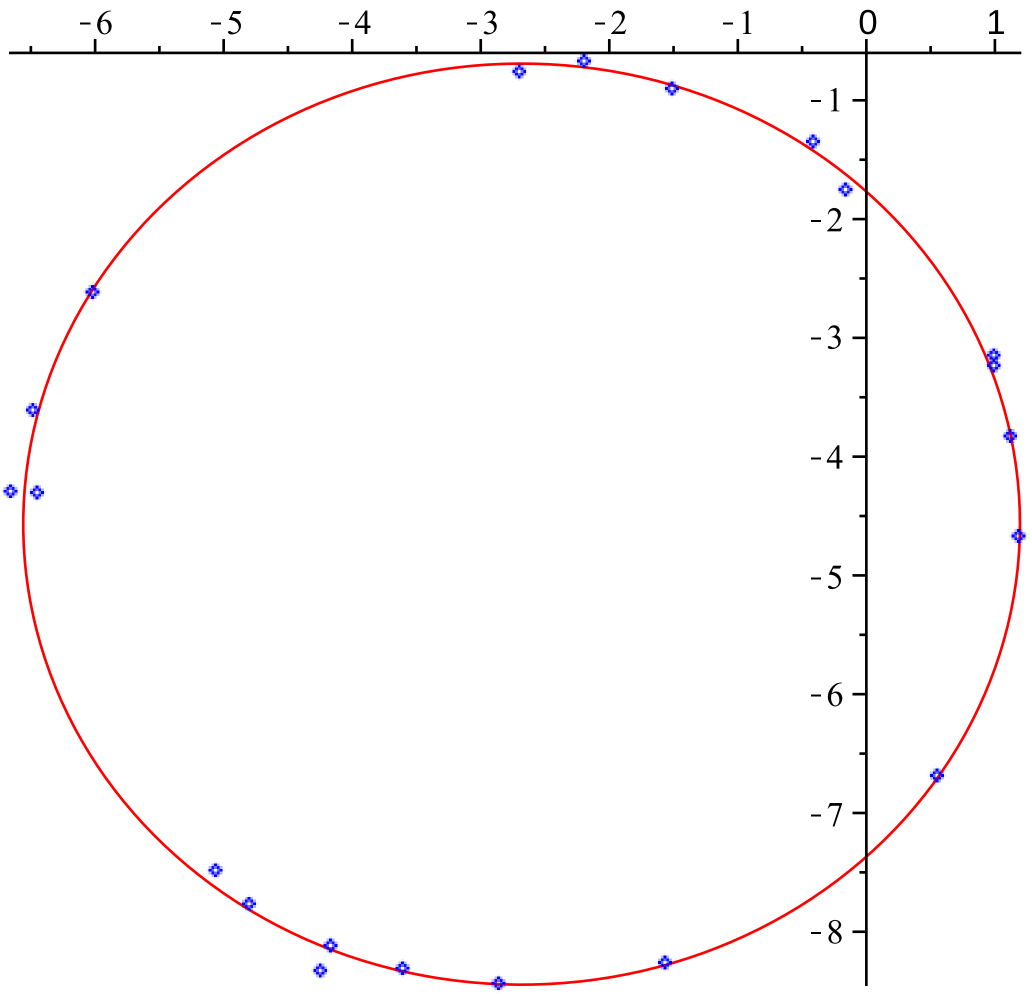
```
> circ:=(a,b,r)-> [ r*cos(t)+a, r*sin(t)+b, t=0..2*Pi];
```

$$circ := (a, b, r) \rightarrow [r\cos(t)+a,\, r\sin(t)+b,\, t=0\,..\,2\,\pi] \qquad \text{(28)}$$

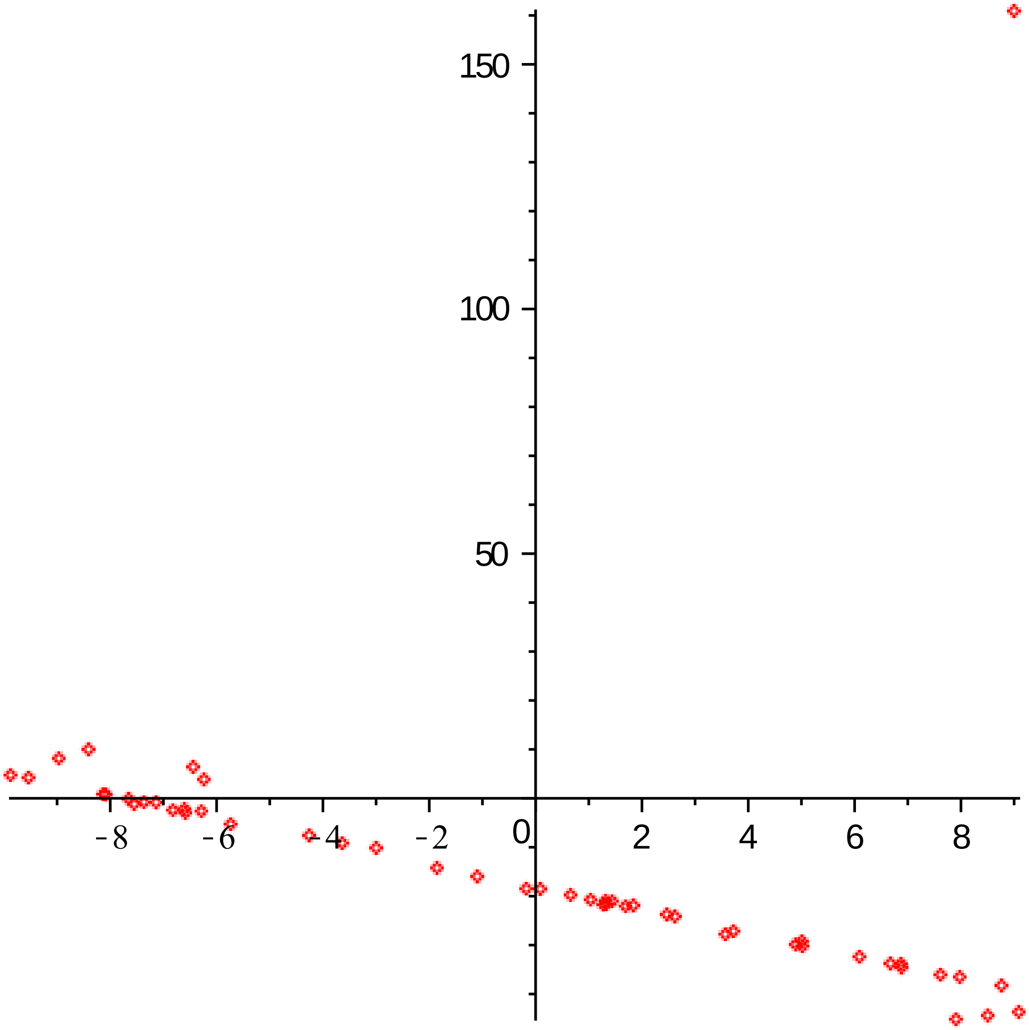
```
> ABR:=subs(op(mine), [a,b,r ] ) ;
```

$$ABR := [-2.682127600,\, -4.568186883,\, 3.877311723] \qquad \text{(29)}$$

```
> plot([circ(op(ABR)),cpts], style=[line,point], color=[red,blue]);
```



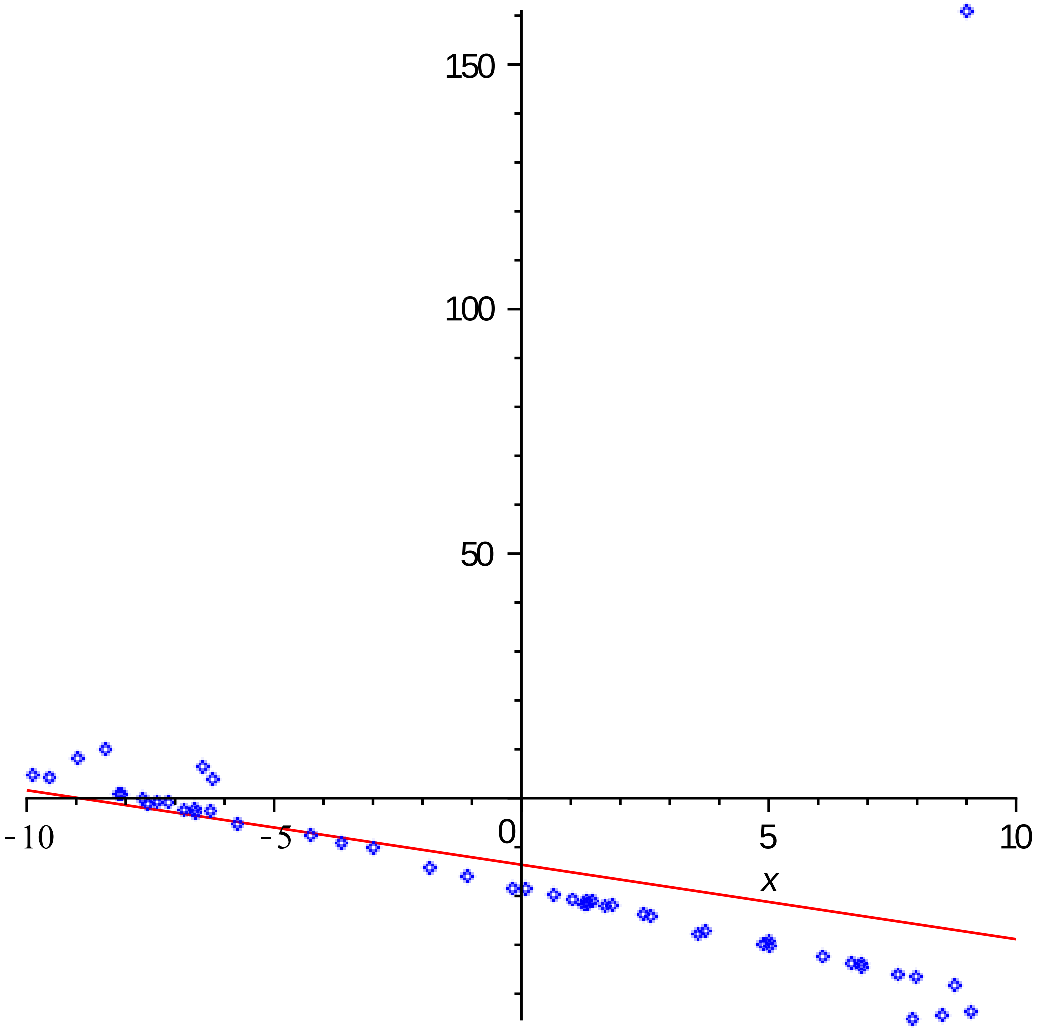
```
> lpt:=bad_line_pts():
  plot(lpt,style=point);
```



```
> my:=CurveFitting[LeastSquares](lpt,x);
```

$$my := -13.6071191857293 - 1.52246845291090\,x \qquad (30)$$

```
> plot([my,lpt],x=-10..10,style=[line,point],color=[red,blue]) ;
```


150
100
50
0
-10
-5
5
10
x