Continuing with graphing

Plotting a function

In [1]:
```
def simple_plot_data(fn, xlow, xhigh, xstep=1, yscale=1, yoffset=0) :
    """This function cooks up data points for a very obnoxious looking gr
aph of the function."""

    retval = []
    x = xlow
    while x < xhigh + xstep :
        y = fn(x)
        retval.append((x, (y + yoffset) * yscale))
        x += xstep
    return retval
```

To try it out :

In [2]:
```
simple_plot_data(lambda x : x*x, 0, 10)
```

Out[2]:
```
[(0, 0),
 (1, 1),
 (2, 4),
 (3, 9),
 (4, 16),
 (5, 25),
 (6, 36),
 (7, 49),
 (8, 64),
 (9, 81),
 (10, 100)]
```

In [3]:
```
simple_plot_data(fn=lambda x : x*x*x, xlow=-10, xhigh=10, xstep=2, yscale
=0.04, yoffset=1000)
```

Out[3]:
```
[(-10, 0.0),
 (-8, 19.52),
 (-6, 31.36),
 (-4, 37.44),
 (-2, 39.68),
 (0, 40.0),
 (2, 40.32),
 (4, 42.56),
 (6, 48.64),
 (8, 60.480000000000004),
```

```
 (10, 80.0)]
```

In [4]:
```
def simple_plot(fn, xlow, xhigh, xstep=1, yscale=1, yoffset=0, debug=0) :
    data_pts = simple_plot_data(fn, xlow, xhigh, xstep, yscale, yoffset)
    if debug > 0 :
        print "simple_plot :",
        print data_pts
    # to print the x-axis
    y_is_0 = yoffset * yscale
    if debug > 0 :
        print "simple_plot : y_is_0 = %d" % y_is_0
    # list to store the lines
    plot_list = []
    for (x, y) in data_pts :
        # String to store the line
        str_line = ""
        for i in range(max(int(y_is_0)+1, int(y)+1)) :
            # The +1 is so that those points do get included
            if i == int(y) or i == y_is_0 :
                str_line += "*"
            else :
                str_line += " "
            if debug > 0 :
                print "simple_plot : %s" % str_line
        plot_list.append(str_line)
    return plot_list
```

Let us test this

In [5]:
```
for line in simple_plot(lambda x : x*x*x, -100, 100, xstep=10, yscale=.00
004,yoffset=1000000) :
    print line
```

```
*                                       *
          *                             *
                   *                    *
                          *             *
                               *        *
                                   *    *
                                     *  *
                                      * *
                                       **
                                       **
                                        *
                                        *
                                        *
                                        **
                                        * *
                                        *    *
                                        *       *
```

```
                                         *            *
                                         *                  *
                                         *                           *
                                         *
       *
```

In [6]:

```python
import math
for line in simple_plot(math.sin, -math.pi, math.pi, math.pi/10, yscale=20, yoffset=1.5) :
    print line
```

```
                             **
                       *      *
                  *           *
             *                *
          *                   *
          *                   *
          *                   *
             *                *
                  *           *
                       *      *
                              *
                              *     *
                              *          *
                              *               *
                              *                  *
                              *                   *
                              *                  *
                              *               *
                              *          *
                              *     *
                              *
```

Plotting loci

Suppose you want to plot the solutions to the equation, say x^2 + y^2 = 1. First we shall convert any such equation to an equation of the form f(x, y) = 0. In this case, f(x, y) = x^2 + y^2 - 1.

In [7]:

```python
screen = []
xlen = 25
ylen = 40
```

In [8]:

```python
def init_screen() :
    """Initiates the screen. One can also use this to clear the screen. One should use this everytime one changes the size of the screen."""
    global screen
    global xlen
```

```
    global ylen

    screen = []
    for i in range(xlen) :
        screen.append([])
        for j in range(ylen) :
            screen[i].append(' ')
        # print screen
        # print "********************************************************
****"
```

In [9]:

```
init_screen()
print screen
```

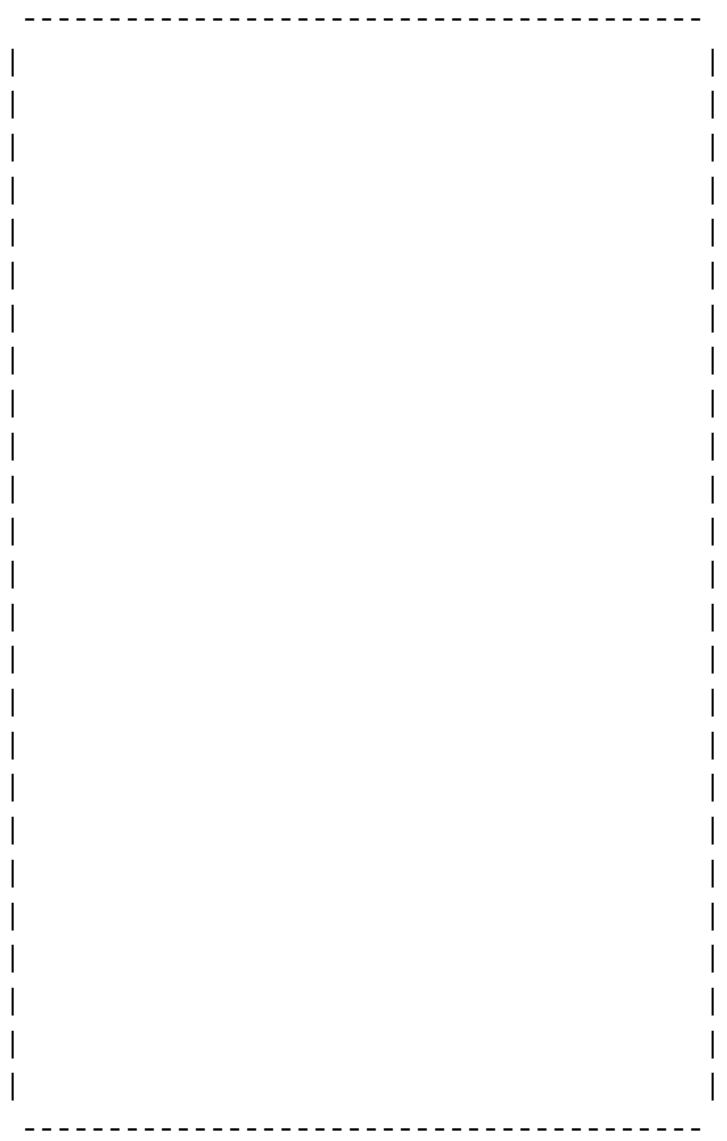
```
[[' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
 ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
 ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
, ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
 ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ', ' ', '
', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
, ' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
 ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' '], [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
 ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
 ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
, ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
 ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ', '
', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
, ' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
 ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' '], [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
 ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
 ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
```

```
, ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
 ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ', '
', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
, ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ', ' ', ' ',
 ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
 ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
 ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
'], [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
, ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
 ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ', '
', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
, ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ', ' ',
 ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
 ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
 ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
', ' '], [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
, ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
 ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', '
', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '
, ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ',
 ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
' ', ' ', ' ', ' ', ' ', ' '], [' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
 ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
 ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', '
', ' ', ' ']]
```

In [10]:

```python
def print_screen() :
    firstline = True
    rule = " "
    for line in screen :
        line_str = "|"
        for char in line :
            line_str += char
            if firstline :
                rule += "-"
        line_str += "|"
        if firstline :
            print rule
            firstline = False
        print line_str
    print rule
```

In [11]:
```
init_screen()
print_screen()
```



In [12]:
```
def print_error(function, warning_type, warning_string) :
    """warning_types : w, e, i"""
    can_cont = 1
    warning_dict = [("w", "Warning"),\
                    ("e", "Error"),\
                    ("i", "Info")]
    warning_fullform = ""
    for (s, term) in warning_dict :
        if s == warning_type :
            warning_fullform = term
    if warning_fullform == "" :
        print "Error : print_error : Invalid error type."
    print "%s : %s : %s" % (warning_fullform, function, warning_string)
    if warning_type == "w" or warning_type == "i" :
        can_cont = 0
    return can_cont
```

In [13]:
```
def plot(x, y) :
```

```
    global screen

    cont = 0
    if x < 0 or x >= xlen :
        cont += print_error(plot, "e", "x out of range")
    if y < 0 or y >= ylen :
        cont += print_error(plot, "e", "y out of range")
    if cont == 0 :
        screen[x][y] = "*"
```

In [14]:

```
init_screen()
for i in range(29) :
    plot(i, 2*i)
print_screen()
```

```
Error : <function plot at 0x1ba0488> : y out of range
Error : <function plot at 0x1ba0488> : y out of range
Error : <function plot at 0x1ba0488> : y out of range
Error : <function plot at 0x1ba0488> : y out of range
Error : <function plot at 0x1ba0488> : y out of range
Error : <function plot at 0x1ba0488> : x out of range
Error : <function plot at 0x1ba0488> : y out of range
Error : <function plot at 0x1ba0488> : x out of range
Error : <function plot at 0x1ba0488> : y out of range
Error : <function plot at 0x1ba0488> : x out of range
Error : <function plot at 0x1ba0488> : y out of range
Error : <function plot at 0x1ba0488> : x out of range
Error : <function plot at 0x1ba0488> : y out of range
 ----------------------------------------
|*                                       |
|  *                                     |
|    *                                   |
|      *                                 |
|        *                               |
|          *                             |
|            *                           |
|              *                         |
|                *                       |
|                  *                     |
|                    *                   |
|                      *                 |
|                        *               |
|                          *             |
|                            *           |
|                              *         |
|                                *       |
|                                  *     |
|                                    *   |
|                                      * |
|                                        |
|                                        |
```

```
|                                       |
|                                       |
|                                       |
 ---------------------------------------
```

In [15]:

```python
def clever_plot(x, y, origx=xlen-1, origy=0) :
    cont = 0
    x = int(x)
    y = int(y)
    array_x = origx - y
    array_y = origy + x
    if array_x < 0 or array_x > xlen - 1 :
        cont += print_error(clever_plot, "e", "x out of range")
    if array_y < 0 or array_y > ylen - 1 :
        cont += print_error(clever_plot, "e", "y out of range")
    if cont == 0 :
        plot(array_x, array_y)
```

In [16]:

```python
for i in range(50) :
    clever_plot(i, i)
print_screen()
```

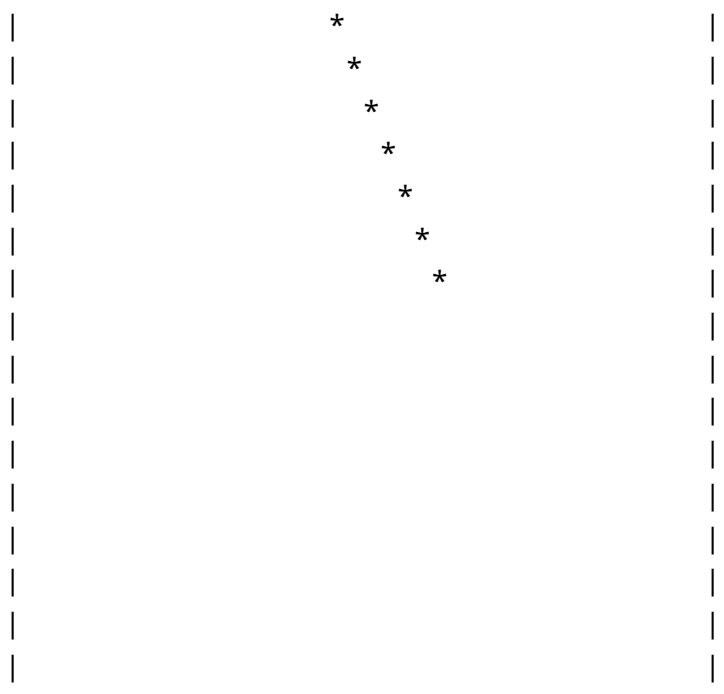
```
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : y out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : y out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : y out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : y out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : y out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : y out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
```

```
Error : <function clever_plot at 0x1ba02a8> : y out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : y out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : y out of range
Error : <function clever_plot at 0x1ba02a8> : x out of range
Error : <function clever_plot at 0x1ba02a8> : y out of range
 ----------------------------------------
|*                        *              |
|  *                    *                |
|    *                *                  |
|      *              *                  |
|        *           *                   |
|          *        *                    |
|            *     *                     |
|              *  *                      |
|                *                       |
|               *  *                     |
|              *     *                   |
|             *        *                 |
|            *           *               |
|           *              *              |
|          *                  *           |
|         *                     *        |
|        *                        *       |
|       *                           *     |
|      *                              *   |
|     *                                 * |
|    *                                    |
|   *                                     |
|  *                                      |
| *                                       |
|*                                        |
 ----------------------------------------
```

In [17]:

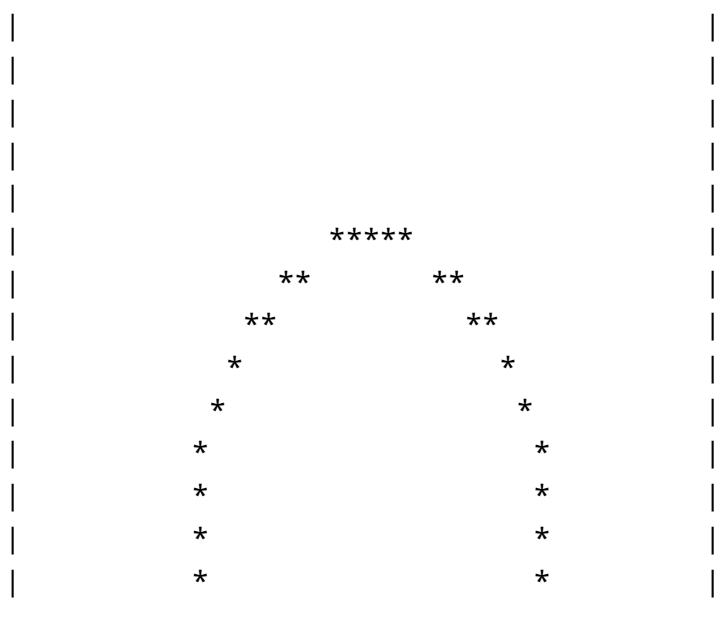
```
init_screen()
for i in range(-5,5) :
    clever_plot(i, 1-i, xlen/2, ylen/2)
print_screen()
```

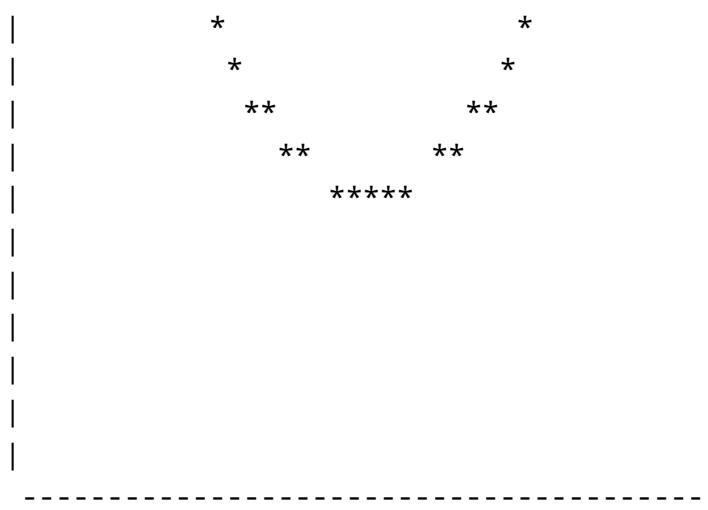
```
 ----------------------------------------
|                                        |
|                                        |
|                                        |
|                                        |
|                                        |
|                                        |
|                                        |
|               *                        |
|                *                       |
|                 *                      |
```



In [18]:
```
def fn_plot(fn, xlow, xhigh, ylow, yhigh, error=.001) :
    init_screen()
    xscale = float(xhigh - xlow) / ylen
    yscale = float(yhigh - ylow) / xlen
    for i in range(ylen) :
        for j in range(xlen) :
            x = (xlow + i * xscale)
            y = (ylow + j * yscale)
            if abs(fn(x, y)) < error :
                clever_plot(i, j)
    print_screen()
```

In [19]:
```
def circle (x, y) :
    return x*x + y*y - 1
fn_plot(circle, -2, 2, -2, 2, .122)
```





In [20]:

```
def parabola (x, y) :
    return (x*x - 4*y)
fn_plot(parabola, -10, 10, -1, 5, .6)
```