EX.0.1.2.d Langou

Let $p(x)$ be the polynomial

$$p(x) = 5x^4 - 2x^3 + 3x^2 - 9x + 7$$

We want to evaluate $p(x)$ at $x = -1$ and $x = 3$.
We will do it using 5 different methods.

1. Given x as `x`, write a simple to expression to compute $p(x)$ the polynomial p evaluated x, as `px`.
2. Given x as `x`, using only statement of the form `z = z + c * x**i` write a sequence of statements that computes `z` such that $z = p(x)$. How many multiplications, additions, and power signs are you using?
3. Given x as `x`, incrementally compute the powers of x with statements of the form `y = y * x` (so that $y = x$, then $y = x^2$, then $y = x^3$, then $y = x^4$, etc.), then using only statement of the form `z = z + c * y`, write a sequence of statements that computes `z` such that $z = p(x)$ by avoiding the `**` power operator. How many multiplications, additions, and power signs are you using?
4. Rewrite the polynomial $p(x)$ in nested form.
5. Given x as `x`, using only statement of the form `y = y * x + c`, write a sequence of statements that computes `y` such that $y = p(x)$. How many multiplications, additions, and power signs are you using?

EX.0.1.2.d, Langou, solution, Langou

See as well the Colaboratory Jupyter Notebook.

1. Given x as `x`, write a simple to expression to compute $p(x)$ the polynomial p evaluated x, as `px`.

```
import numpy as np

x = np.array( [ -1., 3. ])

px = 5. * x**4 - 2. * x**3 + 3. * x**2 - 9. * x + 7.

print( px )
```

```
[ 26. 358.]
```

2. Given x as `x`, using only statement of the form `z = z + c * x**i` write a sequence of statements that computes `z` such that $z = p(x)$. How many multiplications, additions, and power signs are you using?

```
z = 7
z = z - 9 * x
z = z + 3 * x**2
z = z - 2 * x**3
z = z + 5 * x**4
```

We are using 4 additions/substractions, 4 multiplications, and 3 power signs.

3. Given x as `x`, incrementally compute the powers of x with statements of the form `y = y * x` (so that $y = x$, then $y = x^2$, then $y = x^3$, then $y = x^4$, etc.), then using only statement of the form `z = z + c * y`, write a sequence of statements that computes `z` such that $z = p(x)$ by avoiding the `**` power operator. How many multiplications, additions, and power signs are you using?

```
z = 7.

y = x               # y is x (= x^1)
z = z - 9. * y

y = y * x           # y is x^2
z = z + 3. * y

y = y * x           # y is x^3
z = z - 2. * y

y = y * x           # y is x^4
z = z + 5. * y
```

We are using 4 additions/substractions, 7 multiplications, and 0 power signs.

4. Rewrite the polynomial $p(x)$ in nested form.

$$\begin{aligned} p(x) &= 5x^4 - 2x^3 + 3x^2 - x + 7 \\ &= (((5x - 2)x + 3)x - 1)x + 7 \end{aligned}$$

```
px = ( ( ( 5 * x - 2 ) * x + 3 ) * x - 9 ) * x + 7
```

5. Given x as `x`, using only statement of the form `y = y * x + c`, write a sequence of statements that computes `y` such that $y = p(x)$. How many multiplications, additions, and power signs are you using?

```
y = 5
y = y * x - 2
y = y * x + 3
y = y * x - 9
y = y * x + 7
```

We are using 4 additions/substractions, 4 multiplications, and 0 power signs.

4. (python) Using Python, evaluate with and without nested form at $x = -1$ and $x = 3$.

```
import numpy as np

x = np.array( [ -1., 3. ])

px = 5 * x**4 - 2 * x**3 + 3 * x**2 - 9 * x + 7
print( px )

z = 7
z = z - 9 * x
z = z + 3 * x**2
z = z - 2 * x**3
z = z + 5 * x**4
print(z)

px = ( ( ( 5 * x - 2 ) * x + 3 ) * x - 9 ) * x + 7
print( px )

y = 5
y = y * x - 2
y = y * x + 3
y = y * x - 9
y = y * x + 7
print( y )
```

```
[ 26. 358.]
[ 26. 358.]
[ 26. 358.]
[ 26. 358.]
```