El siguiente es un listado de las pruebas mínimas que deberán realizar para a la clase Polinomio.

Polinomio();

1. Comprobar que construye una instancia del Polinomio cero tal como se define en la IVA (aHil() debería generar “0.0X(0)Y(0)”)

Polinomio(const Polinomio& orig);

1. Comprobar que construye la copia del polinomio cero.
2. Comprobar que construye la copia de cualquier otro polinomio diferente de cero con al menos dos términos.

Polinomio(const string& expoli);

Comprobar que construye bien un polinomio si la expoli es:

1."0.0" --> polinomio cero
2."0X(0)Y(0)" --> polinomio cero
3."0.0X(0)Y(0)" --> polinomio cero
4."0.0X(3)Y(2)+0X(2)Y(1)+0X(0)Y(0)" --> polinomio cero
5. "0X(5)Y(4)+0.0X(3)Y(2)+2.5X(0)Y(0)" --> polinomio 2.5X(0)Y(0)
6. "0X(5)Y(4)+2.5X(0)Y(0)" --> polinomio 2.5X(0)Y(0)
7. "6X(5)Y(3)+3X(3)Y(2)+5X(1)Y(1)+3X(0)Y(1)+4X(0)Y(0)" --> polinomio correspondiente distinto de cero que cumple la IVA.
8. "7.5X(5)Y(3)+3.2X(3)Y(2)+5.0X(1)Y(0)+4.2X(0)Y(0)" --> polinomio correspondiente distinto de cero.
9. “1.5X(0)Y(2)-3X(2)Y(1)” → polinomio correspondiente distinto de cero.
10. “3.2X(3)Y(2)+7.5X(5)Y(3)+5.0X(1)Y(0)+4.2+7.5X(5)Y(3)" → polinomio correspondiente distinto de cero.
11. “7.5X(5)Y(3)-2.5X(5)Y(3)+5.0X(1)Y(0)-3.2X(3)Y(2)-4.2+7.5X(5)Y(3)" → polinomio correspondiente distinto de cero.
12. "+4X(0)Y(0)-6X(5)Y(3)+5X(3)Y(2)-3X(0)Y(1)-3X(3)Y(2)" --> polinomio correspondiente distinto de cero.

En todos los casos hay que verificar que el polinomio resultante cumple la IVA (método verInv()).

Igualmente para los operadores se debe comprobar la IVA.

Polinomio& operator+(const Polinomio& p) const;

*this	p	resultado
cero	cero	cero
cero	3.5X(4)Y(3)+4.1X(2)Y(1)+5.2X(0)Y(0)	3.5X(4)Y(3)+4.1X(2)Y(1)+5.2X(0)Y(0)
3.5X(4)Y(3)+4.1X(2)Y(1)+5.2X(0)Y(0)	cero	3.5X(4)Y(3)+4.1X(2)Y(1)+5.2X(0)Y(0)
3.5X(4)Y(3)+4.1X(2)Y(1)+5.2X(0)Y(0)	2X(4)Y(3)+3X(2)y+2Y(1)+1X(0)Y(0)	5.5X(4)Y(3)+7.1X(2)Y(1)+2Y(1)+6.1X(0)Y(0)
2X(4)Y(3)+3X(2)Y(1)+2X(0)Y(1)+1X(0)Y(0)	3.5X(4)Y(3)+4.1X(2)Y(1)+1X(0)Y(1)+5.2X(0)Y(0)	5.5X(4)Y(3)+7.1X(2)Y(1)+3X(0)Y(1)+6.2X(0)Y(0)
2X(4)Y(3)+3X(2)Y(1)+2X(0)Y(1)+1X(0)Y(0)	3.5X(4)Y(3)+1X(0)Y(1)+5.2X(0)Y(0)	5.5X(4)Y(3)+3X(2)Y(1)+3X(0)Y(1)+6.2X(0)Y(0)
3X(2)Y(1)+2X(0)Y(1)+1X(0)Y(0)	3.5X(4)Y(3)+4.1X(2)Y(1)-1X(0)Y(1)+5.2X(0)Y(0)	3.5X(4)Y(3)+7.1X(2)Y(1)+1X(0)Y(1)+6.2X(0)Y(0)
2X(4)Y(3)+2X(0)Y(1)	4.1X(2)Y(1)+5.2X(0)Y(0)	2X(4)Y(3)+4.1X(2)Y(1)+2X(0)Y(1)+5.2X(0)Y(0)
2X(4)Y(3)+3X(2)Y(1)-2X(0)Y(1)+1X(0)Y(0)	3.5X(4)Y(3)+1X(0)Y(1)-5.2X(0)Y(0)	5.5X(4)Y(3)+3X(2)Y(1)-1X(0)Y(1)+6.2X(0)Y(0)
2X(4)Y(3)+2X(0)Y(1)	4.1X(2)Y(1)+5.2X(0)Y(0)	2X(4)Y(3)+4.1X(2)Y(1)+2X(0)Y(1)+5.2X(0)Y(0)

Polinomio& operator-(const Polinomio& p) const;

Pruebas similares al operador suma.

Polinomio& operator*(const Polinomio& p) const;

Comprobar que construye bien un polinomio si *this y p son:

*this	p	resultado
cero	cero	cero
cero	3.5X(4)Y(3)+4.1X(2)Y(1)+5.2X(0)Y(0)	cero
3.5X(4)Y(3)+4.1X(2)Y(1)+5.2X(0)Y(0)	cero	cero
cero	2X(4)Y(3)+3X(2)Y(1)+2X(0)Y(1)+1X(0)Y(0)	cero
2X(4)Y(3)+3X(2)Y(1)+2X(0)Y(1)+1X(0)Y(0)	cero	cero
2X(3)Y(2)+3X(2)Y(1)	3X(0)Y(0)	6X(3)Y(2)+9X(2)Y(1)
-3X(0)Y(0)	2X(3)Y(2)-3X(2)Y(1)	-6X(3)Y(2)+9X(2)Y(1)
2X(3)Y(2)+3X(2)Y(1)	X(3)Y(2)+3X(2)Y(1)+2X(0)Y(0)	2X(6)Y(4)+9X(5)Y(3)+9X(4)Y(2)+4X(3)Y(2)+6X(2)Y(1)
2X(3)Y(2)+3X(2)Y(1)	X(1)Y(0)+3X(0)Y(1)-2.5X(0)Y(0)	2X(4)Y(2)+6X(3)Y(3)-5X(3)Y(2)+5X(3)Y(2)+5X(3)Y(1)+9X(2)Y(2)-7.5X(2)Y(1)
X(1)Y(0)+3X(0)Y(1)-2.5X(0)Y(0)	2X(3)Y(2)+3X(2)Y(1)	2X(4)Y(2)+6X(3)Y(3)-5X(3)Y(2)+5X(3)Y(2)+5X(3)Y(1)+9X(2)Y(2)-7.5X(2)Y(1)
2X(3)Y(2)	5,5X(4)Y(3)-3X(1)Y(1)	11X(7)Y(5)-6X(4)Y(3)
5,5X(4)Y(3)-3X(1)Y(1)	2X(3)Y(2)	11X(7)Y(5)-6X(4)Y(3)

Polinomio& operator/(const Polinomio& p) const;

ESTE OPERADOR SERÁ OPCIONAL. LOS QUE LO HAGAN TENDRÁN PUNTAJE ADICIONAL DE ACUERDO A LA CANTIDAD DE PRUEBAS EXITOSAS QUE DEMUESTREN.

int verGrado();

Comprobar el grado del polinomio si *this es:

*this	resultado
cero	cero
-2	cero
1.0X(1)Y(0)+12.5X(0)Y(0)	uno
1.0X(0)Y(1)+12.5X(0)Y(0)	uno
2X(3)Y(2)+3X(2)y	tres
3.5X(2)Y(4)+4.1X(2)Y(1)+5.2X(0)Y(0)	cuatro

double eval(double X, double y);

Comprobar el valor de la eXpresion si *this, X y y son:

*this	X	y	resultado
cero	0	0	0
2X(3)Y(2)+3X(2)Y(1)	2.5	2	162.5
3.5X(4)Y(3)+4.1X(2)Y(1)+5.2X(0)Y(0)	1	1.5	23.16
1.5X(3)Y(2)-3X(0)Y(1)	0	1.5	0
2.5X(2)Y(0)+3X(0)Y(2)	2.5	0	15.62

string aHil();

Comprobar que construye una hilera a partir de un polinomio cuando:

*this	resultado
cero	“0.0X(0)Y(0)”
-2.0X(0)Y(0)	“-2.0X(0)Y(0)”
2X(3)Y(2)+3X(2)Y(1)	“2.0X(3)Y(2)+3.0X(2)Y(1)”
3.5X(4)Y(3)+4.1X(2)Y(1)+5.2X(0)Y(0)	“3.5X(4)Y(3)+4.1X(2)Y(1)+5.2X(0)Y(0)”
1.5X(3)Y(2)-3X(0)Y(1)	“1.5X(3)Y(2)-3.0X(0)Y(1)”
2.5X(2)Y(0)+3X(0)Y(2)	“2.5X(2)Y(0)+3.0X(0)Y(2)”

bool verInv();

Comprobar que determina bien si se cumple la IVA si:

*this	resultado
cero	true
-2.0X(0)Y(0)	true
2X(3)Y(0)	true
3.5X(4)Y(3)+4.1X(2)Y(2)+5.2X(0)Y(0)+4.0X(0)Y(0)	true