

Un `Polinomio` è un tipo di dato astratto che deve permettere di modellare il concetto matematico di polinomio a coefficienti interi in una sola variabile, la variabile  $x$ . Pertanto un `Polinomio` è univocamente determinato dal suo grado e dai suoi coefficienti, che sono di numero uguale al grado più uno.

--- Metodi invocati nella PRIMA PARTE di `main.cpp`: ---

✓ `Polinomio P(gr, vett);`

Costruttore generico che crea un `Polinomio` di grado  $gr$  i cui coefficienti interi sono contenuti nel vettore `vett`, di lunghezza  $gr + 1$ . In particolare si assuma che `vett[0]` contenga il coefficiente del monomio di grado più alto ( $x^{gr}$ ), `vett[1]` quello di grado immediatamente minore ( $x^{gr-1}$ ), e così via. Il grado del polinomio creato deve essere *esattamente*  $gr$ , quindi il coefficiente del monomio di grado più alto non può essere zero. Notare che, con le specifiche sopra descritte, la costante  $0$  *non* è rappresentabile.

✓ `P.valuta(x);`

Funzione che calcola il valore del polinomio nel punto  $x$ . Si assuma  $x$  intero.

✓ `cout << P;`

Operatore di uscita per oggetti di tipo `Polinomio`. Un polinomio viene stampato secondo il seguente formato:

`-3x^2 +7x^1 -4`

NB: Quantanche uno dei coefficienti sia zero, va comunque stampato il coefficiente  $+0$  davanti al relativo monomio, come nel seguente esempio:

`+5x^3 +0x^2 +0x^1 +7`

In generale tutti i coefficienti positivi debbono essere stampati con il  $+$  davanti. L'operatore *non* aggiunge il *new line* alla fine.

✓ `~Polinomio();`

Distruttore per oggetti di tipo `Polinomio`.

--- Metodi invocati nella SECONDA PARTE di `main.cpp`: ---

✓ `(bool) P;`

Operatore di conversione a booleano per il tipo `Polinomio`, che restituisce `true` nel caso in cui i segni dei coefficienti siano *coerenti*, ossia **o tutti non-negativi** oppure **tutti non-positivi**. In tutti gli altri casi (ossia quando sono presenti contemporaneamente sia coefficienti positivi che negativi) l'operatore deve restituire `false`. Ad esempio per il polinomio `-3x^2 +0x^1 -5` deve restituire `true`, mentre per il polinomio `-3x^2 +0x^1 +5` deve restituire `false`.

✓ `P2 = P;`

Operatore di assegnamento, che assegna `P` a `P2`.

✓ `P*P2;`

Operatore di prodotto tra oggetti di tipo `Polinomio`. Restituisce un nuovo `Polinomio` di grado pari alla somma dei gradi dei due polinomi, e con coefficienti dati dalla definizione matematica di prodotto tra polinomi. Ad esempio il prodotto di `-3x^1 +1` e `+5x^1 -4` deve fornire come risultato il `Polinomio`:

`-15x^2 +17x^1 -4`

Altro esempio, il prodotto di `-3x^1 +1` e `-15x^2 +17x^1 -4` deve fornire il `Polinomio`:

`+45x^3 -66x^2 +29x^1 -4`

---

// MAIN DA UTILIZZARE (si consiglia di fare copia/incolla da questo pdf)

```
#include <iostream>
#include "compito.h"

// ATTENZIONE! Per la soluzione di questo compito e'
// vietato utilizzare la funzione pow di <cmath>

int main() {

    // PRIMA PARTE
    int vett[] = {-3, 7, -4};
    Polinomio P(2, vett); // crea il seguente polinomio di grado 2: P(x) = -3x^2 +7x^1 -4
    cout << "P(x) = " << P << endl << endl;

    cout << "Valuto il polinomio P(x) in x=3: " << endl;
    cout << "P(3) = " << P.valuta(3) << endl << endl; // deve stampare -10
    {
        Polinomio P2(2, vett);
        cout << "Test distruttore:" << endl;
    }
    cout << "distruttore invocato." << endl;

    // SECONDA PARTE
    cout << "Test dell'operatore di conv. a bool ('caso coeff. incoerenti')" << endl << endl;
    if ( P )
        cout << "Il polinomio P(x) ha coefficienti coerenti" << endl << endl;
    else
        cout << "Il polinomio P(x) ha coefficienti incoerenti" << endl << endl; // OK

    int vett2[] = {+2, -3, +5, -1};
    Polinomio P2(3, vett2); // P2(x) = +2x^3 -3x^2 +5x -1
    cout << "P2(x) = " << P2 << endl << endl;

    cout << "Prodotto di P*P2 (deve stampare '-6x^5 +23x^4 -44x^3 +50x^2 -27x^1 +4')" << endl;
    cout << P * P2 << endl << endl;

    P2 = P;
    cout << "Nuovo P2(x) = " << P2 << endl;

}
```

---

#### USCITA CHE DEVE PRODURRE IL PROGRAMMA

P(x) = -3x<sup>2</sup> +7x<sup>1</sup> -4

Valuto il polinomio P(x) in x=3:

P(3) = -10

Test distruttore:  
distruttore invocato.

Test dell'operatore di conv. a bool ('caso coeff. incoerenti')

Il polinomio P(x) ha coefficienti incoerenti

P3(x) = +2x<sup>3</sup> -3x<sup>2</sup> +5x<sup>1</sup> -1

Prodotto di P\*P3 (deve stampare '-6x<sup>5</sup> +23x<sup>4</sup> -44x<sup>3</sup> +50x<sup>2</sup> -27x<sup>1</sup> +4')

-6x<sup>5</sup> +23x<sup>4</sup> -44x<sup>3</sup> +50x<sup>2</sup> -27x<sup>1</sup> +4

Nuovo P3(x) = -3x<sup>2</sup> +7x<sup>1</sup> -4

---