3.14 Progetto di metà corso: Polinomi in una variabile

Un polinomio a coeffecienti interi (ad esempio $2x^4 - 3x^2 + x + 6$ è un polinomio nella variabile x a coefficienti interi) è rappresentato come una lista dinamica di monomi ordinati per esponente decrescente. Il polinomio nullo è rappresentato con la lista vuota.

Vogliamo gestire la memoria tramite la tecnica della condivisione controllata. Quindi, un monomio è rappresentato tramite una classe `Monomio` avente i seguenti campi dati.

`coefficiente` : il coefficiente intero del monomio;

`esponente` : l'esponente del monomio;

`riferimenti` : numero di puntatori che puntano al monomio;

`next` : puntatore smart al monomio successivo.

Come abbiamo visto, un puntatore smart ad un `Monomio` è rappresentato tramite una classe con un unico campo dati di tipo puntatore a `Monomio` (nella classe `Monomio` è presente il campo dati `riferimenti`). Per rendere smart tale classe si ridefiniscono il costruttore di copia, il distruttore e l'assegnazione in modo tale da controllare la condivisione di memoria. Inoltre si ridefiniranno gli operatori `*`, `->`, `==`, `!=`, etc (tutto ciò che risulterà necessario).

Possiamo quindi impostare la classe `Polinomio` nel seguente modo.

```
class Polinomio {
private: // parte privata di Polinomio
  class Monomio; // dichiarazione incompleta
  class SmartP {
  public:
    Monomio* punt;
    ...
  };
  class Monomio {
  public:
    int coefficiente, esponente, riferimenti;
    SmartP next; // puntatore smart
    ...
  };
SmartP first; // smart pointer al primo monomio
public: // parte pubblica di Polinomio
  ...
};
```

La parte pubblica di `Polinomio` dovrà contenere costruttori, metodi e ridefinizioni di operatori in modo tale che il seguente esempio di `main()` compili ed esegui provocando le stampe riportate a commento.

```
int main() {
  Polinomio X(1,1);
  cout << "X = " << X  << endl; // stampa: X = x
  Polinomio Z(3, 4);
  cout << "Z = " << Z  << endl; // stampa: Z = 3x^4
  Polinomio T(X);
  cout << "T = " << T  << endl; // stampa: T = x
  T = Z;
  cout << "T = " << T  << endl; // stampa: T = 3x^4
  cout << "X = " << X << endl; // stampa: X = x
  Polinomio Q=3*X + 2*X*X - X*X*X + 7*(X^4);
  cout << "Q = " << Q << endl; // stampa: Q = 7x^4-x^3-2x^2+3x
  Polinomio P = 2*(X^2);
  cout << "P = " << P << endl; // stampa: P = 2x^2
  cout << "Q*Q = " << Q*Q << endl;
    // stampa: Q*Q = 49x^8-14x^7+29x^6+38x^5-2x^4+12x^3+9x^2
  cout << "Q/P = " << Q/P << endl; // stampa: Q/P = 3x^2+1
  cout << "Q^3 = " << (Q^3) << endl;
    // stampa: Q^3 = 2401x^32-1372x^27+2744x^26+...
  cout << "Q%P = " << Q%P << endl; // stampa: Q/P = x^4-x^3+3x
  cout << "Q(3) = " << Q(3) << endl; // stampa: Q(3) = 567
  if (P == 2*(X^2)) cout << "P(3) = " << P(3) << endl;
    // stampa: P(3) = 18
  if (P != Q) cout << "Q(P(3)) = " << Q(P(3))  << endl;
    // stampa: Q(P(3)) = 729702
}
```

Per realizzare i metodi ricorsivamente è comodo pensare i polinomi $P^n(x)$ definiti ricorsivamente sul grado $n \geq -1$ come segue:

– Il polinomio nullo 0 è l'unico polinomio $P^{-1}(x)$ di grado -1:

$$P^{-1}(x) = 0;$$

– Un polinomio $P^n(x)$ di grado $n \geq 0$ è la somma di un monomio ax^n di grado n e coefficiente $a \neq 0$ con un polinomio $R^k(x)$ di grado $k < n$:

$$P^n(x) = ax^n + R^k(x).$$

Ad esempio la somma $P^n(x)+Q^m(x)$ di due polinomi $P^n(x)$ e $Q^m(x)$ può essere definita (e calcolata) ricorsivamente nel modo seguente:

- Se $P^n(x) = 0$ allora $P^n(x) + Q^m(x) = Q^m(x)$.
- Se $Q^m(x) = 0$ allora $P^n(x) + Q^m(x) = P^n(x)$.

- Altrimenti, cioè quando $n \geq 0$ e $m \geq 0$,

  $$P^n(x) = ax^n + R^k(x) \quad \text{e} \quad Q^m(x) = bx^m + T^h(x)$$

  dove:

  - se $n > m$ allora $P^n(x) + Q^m(x) = ax^n + (R^k(x) + Q^m(x))$;
  - se $n < m$ allora $P^n(x) + Q^m(x) = bx^m + (P^n(x) + T^h(x))$;
  - se $m = n$ e
    * $a + b = 0$ allora $P^n(x) + Q^m(x) = R^k(x) + T^h(x)$;
    * $a + b \neq 0$ allora $P^n(x) + Q^m(x) = (a + b)x^m + (R^k(x) + T^h(x))$.

Volendo usare queste definizioni sarà opportuno definire due metodi privati che estraggono da un polinomio non nullo il primo monomio e il resto del polinomio.