Analisi e progettazione del software
Compito di metà corso
28 novembre 2011

Esercizio 1 (punti 8) Si consideri la classe `Polinomio` vista ad esercitazione, con i campi `max`, grado massimo del polinomio, e `v`, vettore dinamico di `double` di dimensione `max+1` contenente i coefficienti fino al grado `max` incluso.

Si aggiunga alla classe l'operatore `*` che calcola il prodotto tra due polinomi (si assuma per semplicità che nessuno dei due polinomi possa essere nullo).

Ad esempio, il prodotto tra il polinomio $3x^2 - 2x + 4$ e il polinomio $2x^3 - 1$ è il polinomio $6x^5 - 4x^4 + 8x^3 - 3x^2 + 2x - 4$. In altri termini, il programma

```
int main()
{
  Polinomio p1, p2;
  p1 = Polinomio(3,2) + Polinomio(-2,1) + Polinomio(4,0);
  p2 = Polinomio(2,3) + Polinomio(-1,0);
  cout << "Il prodotto tra " << p1 << " e " << p2 << " e': " << endl
       << p1 * p2 << endl;
  return 0;
}
```

dovrà produrre il seguente output

```
Il prodotto tra <+3x^2-2x^1+4x^0> e <+2x^3-1x^0> e':
<+6x^5-4x^4+8x^3-3x^2+2x^1-4x^0>
```

Esercizio 2 (punti 8) Si consideri una stringa che possa contenere unicamente i sei caratteri `(`, `[`, `{`, `)`, `]` e `}`. La stringa si dice *ben formata* se a tutte le parentesi aperte corrispondono parentesi chiuse dello stesso tipo e inoltre due coppie di parentesi non si intrecciano.

Ad esempio, la stringa `"{[[]()]}([])[]"` è ben formata, mentre non lo sono le stringhe `"{[}]"` (parentesi intrecciate), `"([])()[)"` (parentesi quadra chiusa da parentesi tonda), `"([["` (parentesi non chiuse), `")("` (chiusa prima di aperta) e `"())"` (parentesi non aperte).

Si scriva una funzione che riceve un parametro di tipo `string` contenente una stringa siffatta (cioè contenente solo parentesi) e verifica se è ben formata.

Algoritmo risolutivo: Si analizzano i caratteri della stringa uno alla volta a partire dalla posizione 0. Quando si incontra una parentesi aperta, la si inserisce in una pila. Quando si incontra una parentesi chiusa, si estrae un elemento dalla pila (se c'è, altrimenti è un errore) e si verifica che questo sia la corrispondente parentesi aperta al carattere analizzato, altrimenti si ha un errore di tipo di parentesi. Alla fine della lettura della stringa, questa sarà ben formata se la pila è vuota e non ci sono stati errori.

Si assuma di avere disponibile la classe `Pila` con le funzioni `Push()`, `Pop()`, `Top()` ed `EstVuota()`(vista in classe) in cui il tipo base è `char`, invece che `int` come fatto in classe.

Esercizio 3 (punti 17) Si considerino la classe `A` e la funzione `main()` definite qui sotto.

```
class A
{
friend ostream&
 operator<<(ostream& os, const A& a);
public:
  A(unsigned n = 3);
  int& operator[](unsigned i)
     { return v[i]; }
  void BackUp();
  void Restore();
  unsigned Dim() const
    { return dim; }
private:
  unsigned dim;
  int* v;
  int* vb; // vettore di back-up
};

A::A(unsigned n)
{
  unsigned i;
  dim = n;
  v = new int[dim];
  for (i = 0; i < dim; i++)
    v[i] = 1;
  vb = NULL;
}

void A::BackUp()
{
  unsigned i;
  if (vb == NULL)
    vb = new int[dim];
  for (i = 0; i < dim; i++)
    vb[i] = v[i];
}

void A::Restore()
{
  unsigned i;
  assert (vb != NULL);
  for (i = 0; i < dim; i++)
    v[i] = vb[i];
}

ostream& operator<<(ostream& os, const A& a)
{
  unsigned i;
  if (a.vb == NULL)
    for (i = 0; i < a.dim; i++)
      os << a.v[i] << " ";
  else
    for (i = 0; i < a.dim; i++)
      os << a.v[i] << "/" << a.vb[i] << " ";
  return os;
}

int main()
{
  A a1(5), a2, a3();
  a1[3] = 5;
  a1++;
  a2[2] = a1[3] + a1[4] + 2;
  a1.BackUp();
  a1[3]++;
  a1.v[2] = 5;
  a2.Restore();
  cout << a1 << endl << a2 << endl << a3;
  return 0;
}
```

3.1 (3 punti) Segnalare le istruzioni (o parti di esse) della funzione `main()` che danno errore in compilazione o in esecuzione e spiegare brevemente il motivo.

3.2 (3 punti) Riportare cosa stampa il programma una volta eliminate le parti che generano errore.

3.3 (8 punti) Scrivere il costruttore di copia e l'operatore di assegnazione della classe `A` in modo che evitino la condivisione di memoria.

3.4 (3 punti) Scrivere il distruttore della classe `A` che rilasci la memoria non più utilizzata.