Analisi e progettazione del software
Compito di metà corso
26 novembre 2013

Esercizio 1 (punti 4) Si consideri la classe `Pila` come definita a lezione. Si riscriva la funzione membro `Pop()` in modo che la dimensione del vettore venga dimezzata quando la pila occupa meno di un quarto del vettore e il vettore è più grande della sua dimensione iniziale (100).

Esercizio 2 (punti 8) Si consideri la classe `Polinomio`, utilizzando però come rappresentazione un vettore di coppie composte da coefficiente e grado. Si utilizzino le classi `vector` e `pair`, dichiarando quindi come membro privato il seguente unico dato

```
vector<pair<unsigned,float> > vet;
```

Il vettore `vet` deve contenere solo i termini per i quali il coefficiente è diverso da zero, ed essere ordinato per grado crescente.

Per questa classe si definiscano il costruttore con due parametri (che costruisce un monomio), l'operatore `+` e il selettore che, data una potenza qualsiasi, restituisce il suo coefficiente.

Esercizio 3 (punti 8) Un file contiene un insieme di date ordinate cronologicamente (dalla più remota alla più recente). Come esempio, si consideri il seguente file

```
24/10/2013
28/10/2013
1/11/2013
12/11/2013
10/12/2013
```

Si scriva una funzione che prenda un parametro di tipo `string` contenente il nome di un file siffatto e restituisca la data che è la *media* (approssimata per difetto) delle date presenti nel file. Si assuma disponibile la classe `Data` con tutti le funzioni che si ritengono opportuni, inclusi l'operatore `-` che restituisce la differenza in giorni tra due date e l'operatore `+` che somma un numero di giorni ad una data. Nell'esempio, la data media da restituire è il `8/11/2013`.

Suggerimento: Fare la media degli interi ottenuti come differenza tra le date nel file e una data di riferimento, e poi sommarla alla data di riferimento.

Esercizio 4 (punti 4) Data la classe `A` definita come segue (a sinistra la classe, a destra le funzioni mancanti) e dato il programma seguente (funzione `main`):

- individuare tutte le istruzioni che generano errore in compilazione;
- eliminate tali istruzioni, cosa stampa il programma? [motivare le risposte]

```
class A
{
 public:
  A(unsigned u);
  unsigned Size() const { return size; }
  int Get(unsigned a) const;
  void Set(unsigned a, int k);
  void Flip();
 private:
  bool b;
  unsigned size;
  int* p1;
  int* p2;
};
```

```
A::A(unsigned u)
{
  unsigned i;
  size = u;
  p1 = new int[size];
  p2 = NULL;
  b = true;
  for (i = 0; i < size; i++)
    p1[i] = 0;
}

int A::Get(unsigned a) const
{
  if (b) return p1[a];
  else return p2[a];
}
```

```
int main()
{
  A a1(5);
  A a2(5,2);
  a1.Set(1,-4);
  a1.Set(2,3);
  a1.size = 6;
  a1.Flip();
  a1.Size() = 8;
  a1(5);
  a1.Set(1,-7);
  a1.Flip();
  cout << a1.Get(1) << endl;
}
```

```
void A::Set(unsigned a, int k)
{
  if (b) p1[a] = k;
  else p2[a] = k;
}

void A::Flip()
{
   unsigned i;
   b = !b;
   if (p2 == NULL)
   {
     p2 = new int[size];
     for (i = 0; i < size; i++)
       p2[i] = 0;
   }
}
```

Esercizio 5 (punti 8) Dato il programma seguente:

- scrivere che cosa stampa;
- aggiungere il costruttore di copia e l'operatore di assegnazione della classe `A` in modo da non dar luogo a condivisione di memoria;
- scrivere che cosa stampa il programma così modificato. [motivare le risposte]

```
int main()
{
  A a1(5), a2(5);
  a1.Set(3,-4);
  a1.Set(4,9);
  a2 = a1;
  a2.Set(4,-6);
  a2.Flip();
  cout << a1.Get(4) << ' ' << a2.Get(4) << endl;
}
```