

Analisi e progettazione del software

Prova parziale

23 novembre 2022

Esercizio 1 (punti 5) Si consideri la classe `Data` con tutti i metodi e gli operatori visti in classe. Si aggiunga alla classe l'operatore `|` che restituisce la data media tra i due operandi, eventualmente approssimata per eccesso. Ad esempio, se `d1` ha valore 26/10/2022 e `d2` ha valore 4/11/2022, allora `d1|d2` avrà valore 31/10/2022.

Esercizio 2 (punti 13) Un file contiene i risultati delle partite di tennis tra i soci, identificati per semplicità tramite solo il nome (senza spazi). I risultati sono scritti uno per riga, nel formato che si evince dal seguente esempio (si noti che il file non è ordinato).

```
Francesca batte Paolo giocata il 17/11/2022
Paolo batte Giovanni giocata il 20/11/2022
Mario batte Paola giocata il 14/11/2022
Luca batte Paolo giocata il 26/11/2022
Paolo batte Francesca giocata il 16/11/2022
Corrado batte Giulia giocata il 28/11/2022
```

Si scriva una funzione che riceva come parametri il nome di un file siffatto e una stringa contenente il nome di un socio. La funzione deve restituire una coppia contenente il numero di giorni tra la prima e l'ultima partita giocate dal socio e la data della sua ultima sconfitta. Si assuma che il socio abbiamo subito almeno una sconfitta.

Ad esempio, per il file precedente, se il secondo parametro vale `Paolo`, la funzione deve restituire la coppia `<10, 26/10/2022>`.

Si scriva inoltre un breve *driver* che legga i due parametri della funzione a riga di comando e stampi sul monitor il risultato dell'invocazione della funzione.

Esercizio 3 (punti 12) Si considerino la classe A e la funzione main definite di seguito.

```
class A
{public:
    A(unsigned n, int e = 0);
    int operator[]( unsigned i) const
        { return v1[i] + v2[i]; }
    void Set1(unsigned i, int e)
        { v1[i] = e; }
    void Set2(unsigned i, int e)
        { v2[i] = e; }
    void Copy(bool b);
private:
    int* v1;
    vector<int> v2;
};

void A::Copy(bool b)
{ unsigned i;
  if (b)
      for (i = 0; i < v2.size(); i++)
          v1[i] = v2[i];
  else
      for (i = 0; i < v2.size(); i++)
          v2[i] = v1[i];
}
```

```
A::A(unsigned n, int e)
    : v2(n,e)
{ v1 = new int[n];
  for (unsigned i = 0; i < n; i++)
      v1[i] = e;
}

int main ()
{ A a1(4), a2(3,5);
  A a3;
  a2.Set2(2,7);
  a2.Set2(1,3);
  a2.Copy(true);
  a1[0] = 5;
  a1 = a2;
  a1.Set1(2,4);
  a1.Set2(2,2);
  cout << a2[0] << " " << a2[1]
        << " " << a2[2] << endl;
  return 0;
}
```

- Identificare le istruzioni che danno errore il compilazione.
- Riportare cosa stampa il programma dopo aver eliminato tali istruzioni e cosa invece dovrebbe stampare in assenza di interferenza.
- Scrivere il costruttore di copia, l'operatore di assegnazione e il distruttore della classe A in modo da evitare la condivisione di memoria e rilasciare la memoria dinamica non più utilizzata.