Fondamenti di Programmazione 2
(Corso di Laurea in Informatica)
Esame del 15/6/2023

UNIVERSITÀ DELLA CALABRIA

Esercizio 1

Data la seguente porzione di programma rispondere alle domande corrispondenti:

```
#include <iostream>
using namespace std;
void funzione(char& a, char b){
   char c = a;
   a = b+2;
   b = c;
}

int main() {
   int *m = new int[6]{.. la tua matricola ..};

   // 1. Cosa stampa questa istruzione?
   cout << *(m + 2) <<endl;

   // 2. La seguente istruzione è corretta? Se si, cosa stampa?
   cout<< *(m[1]) <<endl;

   // 3. Scegli l'istruzione corretta per gestire la memoria dinamica
   // 3A:
   delete m;
   // 3B:
   delete [] m;
   // 3C:
   non serve deallocare la memoria;
   // 3D:
   for (int i = 0; i < 6; ++i) { delete m[i]; }


   // 4. Cosa stampa questo pezzo di codice?
   char* nome = new char[3]{'a','b','c'};
   funzione(nome[0], nome[1]);
   cout << nome[0] << " " << nome[1] << endl;

}
```

Esercizio 2

Si consideri la seguente classe *Veicolo*

```
class Veicolo {

public:
      Veicolo(string, double, string);
      Veicolo(const Veicolo&);
      string get_targa() const;
      double get_prezzo() const;
      string get_marca() const;

private:
      string targa;
      double prezzo;
      string marca;

}
```

Sfruttando l'ereditarietà, definire opportunamente le classi Auto e Moto. Ciascuna classe dovrà definire il metodo PrezzoFinale che applicherà uno sconto del 25% sul prezzo per l'acquisto di un'auto e del 15% per l'acquisto di una moto. **In tal senso, valutare se è necessario aggiungere nuovi metodi alla classe Veicolo. N.B. Si vuole anche fare in modo che non si possano istanziare oggetti di tipo Veicolo.**

Infine, scrivere un main che: (1) legga da input un elenco di veicoli (Auto e Moto), (2) li memorizzi in un **unico elenco** (scegliere la struttura dati più appropriata, non è consentito costruire un elenco di auto ed un elenco di moto separatamente), (3) li ordini per PrezzoFinale crescente, (4) li stampi in output in modo ordinato.

Fondamenti di Programmazione 2
(Corso di Laurea in Informatica)
Esame del 15/6/2023

UNIVERSITÀ DELLA CALABRIA

NOTA: Per i seguenti Esercizi 3 e 4, si può assumere che il grafo G sia rappresentato da una classe Grafo con la seguente interfaccia (con g un'istanza della classe):

- g.n() restituisce il numero di nodi del grafo
- g(i,j) restituisce true se esiste l'arco diretto tra il nodo i e il nodo j.
- g.w(i) restituisce il peso (di tipo float) del nodo i. ***(Solo Esercizio 3.)***

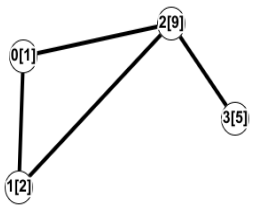
I nodi sono etichettati da 0 a g.n()-1.

Esercizio 3

Scrivere una funzione **esercizio3** che prenda in input un grafo non orientato G e restituisca un arco $\{u, v\}$ per il quale risulta massima la quantità:

$$P(u,v) = P(v,u) = \frac{W(v) \cdot W(u)}{\delta(v) + \delta(v)}$$

dove $W(v)$ denota il peso associato al nodo v e $\delta(v)$ il suo grado **(per ottenere il peso di un nodo si può utilizzare il metodo G.w(i) descritto sopra, mentre non si può assumere esista un metodo nella classe** Grafo **che calcoli** $\delta(v)$**)**.

Esempio: Supponiamo G sia il grafo in figura, con $W(0) = 1, W(1) = 2, W(2) = 9, W(3) = 5$. In questo caso, $P(0,1) = \frac{1\cdot 2}{2+2} = \frac{1}{2}, P(0,2) = \frac{1\cdot 9}{2+3} = \frac{9}{5}, P(1,2) = \frac{2\cdot 9}{2+3} = \frac{18}{5}$ e $P(2,3) = \frac{9\cdot 5}{3+1} = \frac{45}{4}$. Dunque, l'arco per il quale P risulta essere massima è $\{2,3\}$.	

Esercizio 4

Scrivere una funzione **esercizio4** che presi in input un grafo orientato $G(V, E)$ di n nodi e un intero k, restituisca true, se e solo se esiste un cammino semplice (cioè nel quale ogni nodo appare al più una volta) di lunghezza esattamente k che connette il nodo 0 al nodo $n - 1$.

Esempio: Supponiamo G sia il grafo in figura, con $n = 4$ e $k = 3$. In questo caso la funzione restituirà true in quanto è possibile raggiungere il nodo 3 dal nodo 0 attraverso un cammino semplice lungo 3. In particolare, il cammino è $(0, 2, 1, 3)$.