

## Esercizio 1

Data la seguente porzione di programma, rispondere alle domande corrispondenti:

```
string* nc = new string[2]{il tuo nome, il tuo cognome};  
//1: La seguente istruzione è corretta? Se sì, cosa stampa?  
    cout << nc[0][2];  
//2: La seguente istruzione è corretta? Se sì, cosa stampa?  
    cout << *(nc + 1);  
//3: La seguente porzione di codice è corretta? Se sì, cosa fa?  
for (int i = 0; i < nc[0].length(); i++)  
    cout << nc[0][nc[0].length() - 1 - i];  
//4: Come andrebbe deallocata la memoria dinamica allocata in questo main?  
A: delete nc[2];  
B: delete nc;  
C: delete [] nc;  
D: nel main non serve il distruttore.
```

## Esercizio 2

Si implementi una classe astratta Persona che abbia i seguenti campi privati:

string nome, string cognome.

La classe Persona, oltre al costruttore senza parametri, al costruttore con due parametri e ai metodi get e set, deve contenere una funzione **virtuale pura** *int getStipendio()*.

Implementare, successivamente, due sottoclassi Studente e Impiegato che ereditino opportunamente dalla classe Persona e implementino il metodo getStipendio nel seguente modo:

1. La classe Studente deve restituire il valore 10;
2. La classe Impiegato deve restituire il valore 100.

Si noti che, nell'esercizio, si sta assumendo, per semplicità, che ogni studente abbia uno stipendio pari a 10 euro ed ogni impiegato pari a 100 euro.

Implementare, infine, una classe GestorePersone che abbia un campo privato di tipo *vector<Persona\*> persone* e implementi le seguenti funzioni:

1. *void aggiungiPersona(Persona\* p)* che aggiunga p nel vettore delle persone;
2. *double getStipendioMedio()* che restituisca la media degli stipendi delle persone presenti nel vettore delle persone.

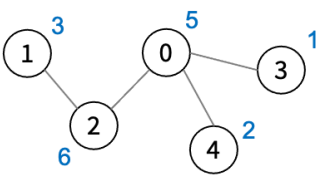
### Esercizio 3

Scrivere una funzione **esercizio3** che prenda in input un grafo non orientato  $G$ , dove ogni nodo in  $G$  ha associato un valore intero positivo chiamato potenziale. Inoltre, per ogni nodo  $v$  si definisce il suo stato come 1 se il potenziale di  $v$  è strettamente maggiore della media del potenziale dei suoi nodi direttamente adiacenti, altrimenti lo stato si definisce come 0. La funzione deve ritornare il nodo con stato pari a 1 e con il potenziale più alto.

Il grafo è rappresentato da una classe Grafo con la seguente interfaccia (con  $g$  un'istanza della classe):

- $g.n()$  restituisce il numero di nodi del grafo,
- $g.m()$  restituisce il numero di archi del grafo,
- $g(i,j)$  restituisce true se esiste un arco tra il nodo  $i$  e il nodo  $j$ .
- $g.pot(i)$  restituisce il potenziale del nodo  $i$ .

I nodi sono etichettati da 0 a  $g.n()-1$ . Se esistono due nodi con lo stesso potenziale più alto e lo stato pari a 1, restituire il nodo con l'indice minore.

<p><i>Esempio:</i> in questa immagine, il potenziale di ogni nodo è indicato dal valore in blu vicino al nodo. Ad esempio, il nodo 0 ha potenziale pari a 5. Lo stato di ogni nodo è il seguente:</p> <ul style="list-style-type: none"><li>- nodo 0, potenziale 5, stato 1 (poiché il potenziale, 5, è strettamente maggiore della media del potenziale dei nodi adiacenti <math>(6+1+2) / 3 = 3</math>)</li><li>- nodo 1, potenziale 3, stato 0</li><li>- nodo 2, potenziale 6, stato 1</li><li>- nodo 3, potenziale 1, stato 0</li><li>- nodo 4, potenziale 2, stato 0</li></ul> <p>La funzione in questo caso restituisce 2, poiché il nodo 2 è il nodo con stato 1 e potenziale più alto tra tutti i nodi con stato 1.</p>	
---	--

### Esercizio 4

Scrivere una funzione **esercizio4** che prenda in input una collezione  $S$  di stringhe, e restituisca true se è possibile suddividere  $S$  in due insiemi disgiunti  $S_1$  ed  $S_2$  tale che la somma dei valori delle stringhe nel primo insieme sia uguale alla somma dei valori delle stringhe nel secondo insieme; in caso contrario, la funzione restituisce false. Il valore di una stringa  $s$  è un numero intero (positivo o negativo) dato da una funzione  $val(s)$ , la quale si può supporre essere globale e già implementata.

Si può assumere che:

- $S$  è rappresentato da un `vector<string>` (`ArrayList<String>` se si usa Java),
- $S$  contiene almeno due stringhe e si può assumere che non contenga più volte la stessa stringa,
- $val$  è una funzione con visibilità globale, prende in input una stringa presente in  $S$ , e restituisce il suo valore (un numero intero).

<p><i>Esempio:</i> la funzione dovrà restituire true poiché è possibile suddividere <math>S</math> in due insiemi tale che la somma dei valori delle stringhe (dati dalla funzione <math>val</math>) sia uguale tra i due insiemi.</p> <p>In questo caso, una possibile suddivisione è la seguente: <math>S_1 = \{ab, q, 8a8b\}</math> e <math>S_2 = \{cde, xyz, q42, ccc\}</math>, dove la somma per <math>S_1</math> è pari a 53, ed è uguale alla somma per <math>S_2</math>.</p>	<p><math>S = \{ab, cde, xyz, q, q42, 8a8b, ccc\}</math> <math>val(ab) = 11</math> <math>val(cde) = 11</math> <math>val(xyz) = 12</math> <math>val(q) = 8</math> <math>val(q42) = 7</math> <math>val(8a8b) = 34</math> <math>val(ccc) = 23</math></p>
--	--