Cognome e Nome _____

Matricola _____

Programmazione e Algoritmi 2 A.A. 2024/2025

Appello del 12 giugno 2025

Esercizio 1 [6 punti]

Si consideri il seguente codice Python.

```
class A:
    def __init__ (self, n:int, m:int):
        self._n= n + m

    @property
    def n (self) -> int:
        return self._n + 1

class B:
    _n:int = 0
    @property
    def n (self) -> int:
        return self._n * 2
```

```
class C (A, B):
    pass

class D (B, A):
    pass

class E (D, A):
    pass

c = C(5, 2)
print ("C " + str(c.n))

e = E(2, 3)
print ("E " + str(e.n))
```

Denotando la classe `object` con O, si calcoli il MRO di tutte le classi, e si scriva in tabella il procedimento e il risultato finale:

classe	**MRO**
A	L(A) = A + merge (L(O), O) = A + merge (O, O) = AO
B	
C	
D	
E	

Si scriva nel riquadro a destra cosa **esclusivamente** ciò che viene stampato a video dal codice.

Esercizio 2 [6 punti]

Si considerino le seguenti sequenze di numeri interi:

A = 6, 11, 21, 14, 15, 34, 28, 4, 17 **B = 21, 28** **C = 2, 8, 19**

1. **[3 punti]** Mostrare **passo-passo l'evoluzione della tabella Hash di dimensione 13 con liste di trabocco** che si ottiene, a partire dalla tabella vuota, inserendo le chiavi della sequenza **A** nell'ordine indicato, rimuovendo quindi le chiavi della sequenza **B** nell'ordine indicato ed inserendo infine le chiavi della sequenza **C** sempre nell'ordine indicato.
2. **[3 punti]** Ripetere l'esercizio utilizzando una **tabella Hash di dimensione 13 con hashing interno (hashing doppio)**, tenendo presente che la seconda funzione Hash dell'hashing doppio deve essere modulo 12.

Esercizio 3 [11 punti]

Tutto il codice richiesto deve essere in **Python** ed annotato con i tipi opportuni.

Si vogliono gestire i movimenti associati al credito di SIM di telefonia mobile.

Si assuma, senza scrivere nulla, l'esistenza di una classe astratta **Movimento** con

- una variabile di istanza **idSim** (una stringa), che contiene il numero telefonico associato alla SIM a cui il movimento si riferisce;
- una variabile di istanza **tempo** (un intero), che rappresenta l'istante del movimento espresso in secondi trascorsi dalle ore 00:00 del 1° gennaio 2000.

e i seguenti metodi di istanza:

- **`__init__`** che inizializza un oggetto della classe **Movimento** assegnando **idSim** e **tempo**;
- Proprietà **`idSim`** e **`tempo`** (in lettura)
- metodo astratto **`get_importo(self) -> float`** che restituisce l'importo in Euro del movimento. In particolare, se il movimento è un movimento di spesa viene restituito un valore minore o uguale a zero (ogni sms costa 0,25 €, mentre ogni chiamata costa 0,10€ di scatto alla risposta più mezzo centesimo per ogni secondo di durata), mentre se è un movimento di ricarica credito viene restituito un valore positivo.

La classe astratta **Movimento** è estesa dalle sottoclassi (non astratte) **Chiamata**, **Sms** e **Ricarica.**

Si tenga conto del fatto che la classe **Chiamata** ha:

- una variabile di istanza **durata** (un intero) che contiene la durata in secondi della chiamata;
- una variabile di istanza **destinatario** (una stringa), che contiene il numero chiamato.

La classe **Sms** deve avere:

- una variabile di istanza **destinatario** (una stringa), che contiene il destinatario dell'SMS.

La classe **Ricarica** deve avere:

- una variabile di istanza **importo** (un float), che contiene l'importo della ricarica e deve valere almeno 0,01.

Si assuma, senza scrivere nulla, che tutte le sottoclassi abbiano i metodi **`__init__`** (che prendono in input gli opportuni parametri e che richiamino opportunamente il metodo **`__init__`** della classe **Movimento**), e proprietà di accesso in lettura per tutti i campi.

[4 punti] Si scriva l'implementazione del metodo **`get_importo(self) -> float`** per tutte le sottoclassi **Chiamata**, **Sms** e **Ricarica** di Movimento.

[7 punti] Si scriva una classe **Gestore** con

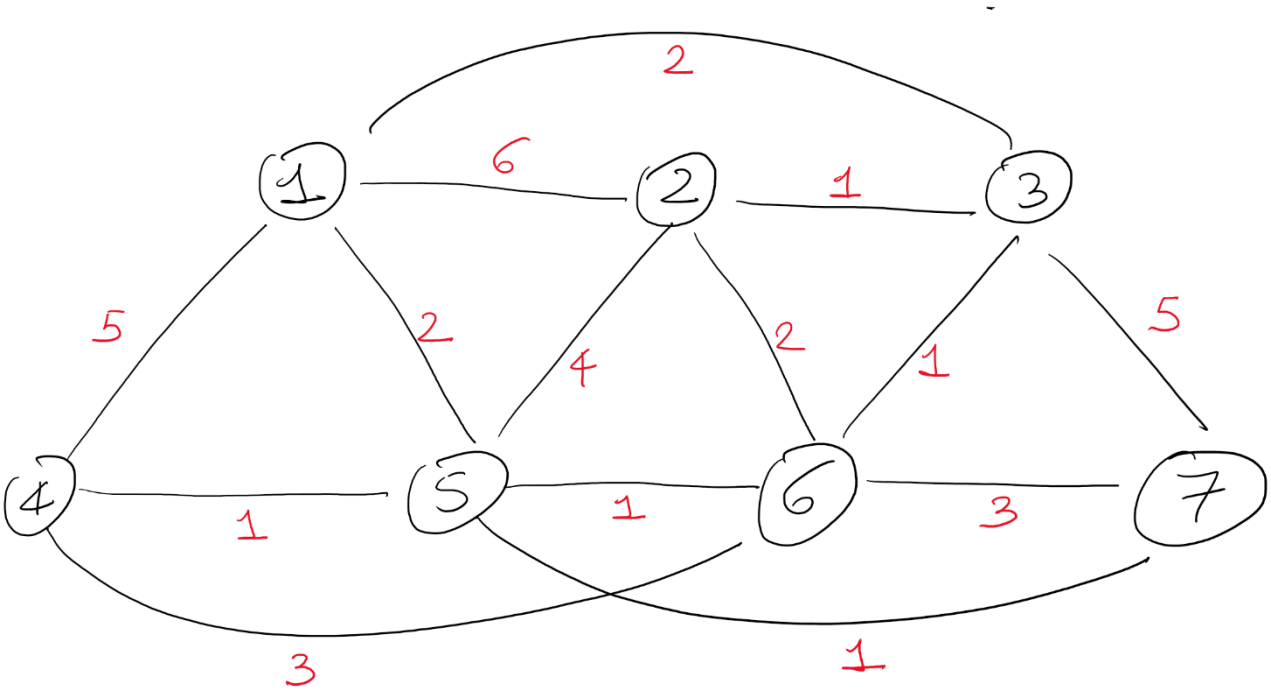
- una variabile di istanza **movimenti** (list di Movimento);

Implementare i seguenti metodi di istanza:

- **`__init__`** che inizializza un oggetto della classe **Gestore** con lista **movimenti** vuota.
- un metodo **`get_credito (self, idSim:string) -> float`** che restituisce il credito presente per la sim **`idSim`** (ottenuto sommando tutti gli importi dei movimenti, di ricarica e di spesa, nella lista **movimenti** relativi alla sim **`idSim`**).
- un metodo **`numero_SMS (self, idSim:string, inizio:int, fine:int) -> int`** che restituisce il numero di sms inviati dalla sim **idSim** nell'intervallo di tempo compreso tra **inizio** e **fine** (estremi inclusi).

Esercizio 4 [11 punti]

Si consideri il grafo diretto in figura.



a. **[5 punti]** Mostrare una possibile esecuzione **passo-passo** dell'algoritmo di **Dijkstra** per i cammini minimi da singola sorgente, a partire dal nodo sorgente **7**. Ad ogni passo, bisogna mostrare il contenuto della coda con priorità utilizzata dall'algoritmo e del vettore dei padri.
Si descriva il cammino minimo dal nodo 7 al nodo 1. Che peso ha?

b. **[5 punti]** Mostrare una possibile esecuzione **passo-passo** dell'algoritmo **Bellman-Ford** per i cammini minimi da singola sorgente, a partire dal nodo sorgente **2**. Mostrare, per ogni iterazione completa che si fa sugli archi, l'evoluzione del **vettore delle distanze** e del **vettore dei padri**, assumendo di visitare, in ogni iterazione, gli archi in ordine crescente rispetto alle etichette dei nodi da cui escono e, a parità di nodo di partenza, in ordine crescente rispetto all'etichetta del nodo di arrivo.
Si descriva il cammino minimo dal nodo 2 al nodo 7. Che peso ha?

c. **[1 punto]** Si dica quale algoritmo, tra **Dijkstra** e **Bellmann-Ford**, è più conveniente utilizzare per risolvere il problema dei cammini minimi da singola sorgente sul grafo dato in figura.

Regole per lo svolgimento della prova scritta:

- Per svolgere il compito si hanno a disposizione **150** minuti.
- Scrivere **subito** nome, cognome, matricola su OGNI FOGLIO (**compreso questo**).
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- È possibile consultare appunti, libri e dispense.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc…) deve essere **completamente disattivato** e **depositato in vista sulla cattedra**
- Mettere in vista sul banco un valido documento di identità.