

Esercizio 1. Implementare una classe StudenteDinamico i cui dati siano (almeno):

- Un intero rappresentante il numero di matricola identificativo dello studente
- Una data rappresentante la data di nascita (si implementi a questo scopo la classe Data)
- Un array **dinamico** di 25 valori interi compresi tra 0 e 30, che rappresenta l'elenco dei voti che lo studente ha preso in ciascuno dei 25 esami previsti dal suo corso di laurea:
 - Voti da 1 a 17 per esami sostenuti ma non superati
 - Voti da 18 a 30 per esami sostenuti e superati
 - 0 per esami mai sostenuti

Dotare la classe dei seguenti metodi (almeno):

- Un costruttore senza parametri e un costruttore di copia
- Distruttore
- Set e get per il numero di matricola
- set e get per il voto di un dato esame
- set e get per la data di nascita
- operatori input/output che consentono di leggere e stampare uno studente
- getMedia che restituisce la media dei voti dello studente
- getVotoMax che restituisce il voto più alto dello studente
- getNumEsamiAllaLaurea che restituisce il numero di esami che mancano allo studente per completare il percorso di studi
- bool studentePiuGiovaneDi(const Studente & S) che restituisce true se lo studente è più giovane dello studente S, false altrimenti.
- Operatore = di assegnamento

Implementare quindi un main che letto in input un array di N studenti, determini e stampi:

- La media generale degli studenti;
- Lo studente più brillante (quello con la media migliore);
- Il voto più alto più frequente;
- Il mese in cui sono nati meno studenti.

Esercizio2. Estendere il progetto precedente implementando una classe Universita che funga da contenitore per gli studenti. I dati membro devono includere (almeno):

- Una stringa rappresentante il nome dell'università.
- Un **array dinamico** di oggetti StudenteDinamico.
- Un intero che tenga traccia del numero attuale di studenti iscritti.

Dotare la classe dei seguenti metodi (almeno):

- Costruttori opportuni
- Distruttore
- Operatore di assegnamento
- aggiungiStudente(const StudenteDinamico& s): inserisce in coda uno studente nell'array degli studenti
- cercaStudente(int matricola): restituisce l'indice dello studente nell'array se presente, -1 altrimenti.
- rimuoviStudente(int matricola): cerca lo studente tramite matricola e, se presente, lo rimuove dall'array facendo attenzione a non lasciare buchi nella struttura.

- `getStudente(int matricola)`: restituisce uno studente cercato (gestire opportunamente il caso di matricola inesistente).
- `setStudente(int matricola, const StudenteDinamico& s)`: sovrascrive i dati dello studente avente quella matricola con i nuovi dati passati.
- `migliorStudente()`: restituisce lo `StudenteDinamico` con la media voti più alta tra tutti gli iscritti.
- `operator<<`: per stampare il nome dell'università e l'elenco completo degli studenti iscritti.

Main: Implementare un main che:

1. Crei un oggetto `Universita`.
2. Permetta all'utente di inserire manualmente nuovi studenti nell'università (utilizzando il metodo `aggiungiStudente`).
3. Esegua una prova di rimozione di uno studente specificando una matricola.
4. Stampi a video l'intero contenuto dell'università tramite `operator<<`.
5. Identifichi e stampi i dati dello studente migliore dell'ateneo.