

Un archivio contiene per ogni lettera dell'alfabeto (si consideri l'alfabeto inglese)  $N$  elementi. Ogni elemento contiene il cognome (di dimensione fissa uguale a 20 caratteri), il nome (di dimensione fissa uguale a 20 caratteri) e la matricola (memorizzata come numero intero) di un dipendente. Si supponga che la prima lettera del cognome e del nome siano maiuscole. Il dipendente è inserito in corrispondenza della lettera con cui inizia il suo cognome. Gli elementi sono ordinati per ordine alfabetico rispetto al cognome.

Le operazioni che possono essere effettuate sul tipo `Archivio` sono le seguenti:

- **Archivio a(N)**  
Costruttore che inizializza l'archivio in modo tale che per ogni lettera dell'alfabeto risultino  $N$  elementi;
- **~Archivio()**  
Il distruttore che libera tutta la memoria occupata dall'archivio;
- **a.inserisci(cg, nm, mat)**  
Operazione che inserisce il dipendente di cognome **cg**, nome **nm** e matricola **mat** nell'archivio; la funzione restituisce **true** se l'inserimento può essere effettuato e se non esiste già un dipendente con lo stesso cognome e la stessa matricola; **false** altrimenti;
- **cout << a**  
L'operatore di uscita per il tipo **Archivio**. I dipendenti vengono stampati in ordine alfabetico. Il formato dell'uscita è il seguente:  
A  
Albini Franco 888888  
Alfieri Piero 990090  
B  
Barga Giacomo 102333  
F  
Fazio Franco 789777
- **Archivio a(a1)**  
Costruttore di copia;
- **a.cancella(cg)**  
Operazione che cancella il dipendente con cognome **cg**. La funzione restituisce **false** se il dipendente non è presente nell'agenda; altrimenti restituisce **true**;
- **a1+=a2**  
Operazione che unisce l'archivio **a2** all'archivio **a1**. Nell'archivio risultante **a1** i dipendenti devono risultare in ordine alfabetico. Se, per una lettera, il numero di elementi necessari per eseguire l'unione è superiore al numero di elementi disponibili per quella lettera in **a1**, l'unione non viene effettuata.

Utilizzando il linguaggio C++, realizzare il tipo di dati astratti **Archivio**, definito dalle precedenti specifiche.

## NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA:

- Effettuare il **login**  
Nome: studenti  
Password: studenti
- Aprire il *Dev-C++* (dal Menù *Avvio* (o *Start*) nella barra degli strumenti in fondo allo schermo, selezionare *Programmi* e quindi *Dev-C++*)
- **Prima di iniziare a svolgere l'elaborato**, selezionare la voce *Identifica studente* nel menù *Strumenti* all'interno dell'ambiente *Dev-C++* e inserire i dati richiesti
- Dal menu *File* del *Dev-C++*, aprire il progetto *esainf.dev* presente nella cartella *c:\esame\esaInf*. Il progetto contiene tre file, denominati *compito.h*, *compito.cpp* e *main.cpp*
- Scrivere la dichiarazione della classe nel file *compito.h* e la definizione delle funzioni nel file *compito.cpp*. Il file *main.cpp* contiene la funzione principale *main()* che serve a verificare le funzioni scritte nel file *compito.cpp*. Il file *main.cpp* può essere modificato. **Si tenga presente, comunque, che in sede di valutazione dell'elaborato verrà considerato esclusivamente il contenuto dei file *compito.h* e *compito.cpp***
- **Per una corretta stampa dell'elaborato**, non scrivere righe di codice di lunghezza eccessiva (mantenersi entro i margini imposti dall'ambiente *Dev-C++* cioè entro la linea verticale presente alla destra della pagina);
- **Per la Consegna:**
  - Selezionare la voce *Consegna* dal menù *Strumenti* (o *Tools*) all'interno dell'ambiente *Dev-C++* e premere il tasto INVIO fino a quando non viene chiusa la finestra che è stata attivata.
  - Aspettare al proprio posto di essere chiamati per verificare la stampa del proprio elaborato, firmarlo e **consegnarlo definitivamente**.

**Condizione necessaria** per la correzione dell'elaborato è che le prime tre funzioni siano state implementate correttamente.