

Un sistema semplificato di gestione di un'azienda può essere realizzato utilizzando le strutture seguenti:

```
struct dipendente
{
    char cognome [LUNG]; int codice; int anni; char categ [LUNG];
};
struct azienda
{
    dipendente dip[NUM]; int numDip;
};
```

con **NUM** il numero massimo di dipendenti dell'azienda e **numDip** il numero di dipendenti effettivamente presenti nell'azienda. Ogni dipendente è individuato dal suo cognome (al massimo di **LUNG** caratteri), da un codice, dagli anni di servizio prestati in azienda e dalla categoria a cui il dipendente appartiene. Quando un'istanza dell'azienda è creata, il valore di **numDip** è inizializzato a zero.

Scrivere il corpo delle seguenti funzioni C++.

1. **bool build(const char nome[], azienda* az)** che utilizza il contenuto del file il cui nome è passato come primo argomento attuale per inizializzare l'oggetto di tipo **azienda** puntato da **az**; il file contiene un numero intero che rappresenta il numero di righe contenute nel file: ogni riga è costituita da una stringa, due numeri interi ed una stringa che rappresentano, rispettivamente, il cognome, il codice, gli anni di servizio e la categoria di un dipendente. Un esempio di riga è il seguente:

```
Rossi 1 10 A
```

dove **Rossi** è il cognome, **1** è il codice che lo identifica nell'azienda, **10** sono gli anni di servizio e **A** è la categoria. La funzione restituisce **false** se il numero di dipendenti supera la dimensione massima del vettore **dip** e se occorrono errori nella lettura del file; altrimenti restituisce **true**;

2. **void ordina(azienda* az)** che ordina i dipendenti per categorie alfabeticamente crescenti e all'interno di ogni categoria per valori alfabeticamente crescenti del campo **cognome**;
3. **void salario(azienda* az, const char categ [], float base)** stampa il salario di ogni dipendente appartenente alla categoria passata come secondo argomento alla funzione: il salario è calcolato come somma del salario base (terzo argomento della funzione) e del 10% del salario base per ogni 5 anni di servizio;
4. **bool prePenz(azienda* az, const char categ [])** che elimina il dipendente con più anni di servizio appartenente alla categoria passata come secondo argomento alla funzione e ricompatta il vettore **dip**; la funzione restituisce **false** se non esistono dipendenti della categoria, altrimenti restituisce **true**;
5. **int numDip(azienda* az, char c)** che restituisce il numero di dipendenti il cui cognome inizia con il carattere passato come secondo argomento alla funzione.

NOTE SULLO SVOLGIMENTO DELLA PROVA:

- PRIMA DI INIZIARE LO SVOLGIMENTO DELL'ELABORATO, POSIZIONARSI nel directory **c:\esame\esainf** ed eseguire il programma **esame.exe**, rispondendo alle domande;
- Per svolgere l'elaborato, lanciare il DevC++, aprire il progetto *esaInf.dev* presente nel directory *c:\esame\esaInf* e scrivere le funzioni richieste nel file *compito.cpp*, già presente nel progetto;
- per una corretta stampa dell'elaborato bisogna mantenere il codice entro i margini imposti dall'ambiente Dev-C++ (linea verticale presente alla destra della pagina);
- per la produzione dei risultati, aprire una finestra DOS, digitare **cd c:\esame\esaInf** e premere il tasto INVIO; quindi digitare **esaInf.exe > output.txt** e premere più volte il tasto INVIO finché non compare di nuovo il prompt del DOS.