

Un sistema di gestione delle CPU per un negozio che vende computer può essere realizzato utilizzando le strutture C++ seguenti:

```
struct CPU
{
    char prod[LUN];
    char ident[LUN];
    float costo;
    int quanti;
};
struct gestCPU
{
    CPU vettCPU[DIM];
    int numCPU;
};
```

dove **DIM** è il numero massimo di CPU gestibili dal sistema e **numCPU** è il numero di CPU correntemente gestite dal sistema. Una CPU è rappresentata da una struttura i cui membri memorizzano il produttore, l'identificatore della CPU, il costo e la quantità disponibile nel negozio. Quando è creata un'istanza di tipo **gestCPU**, **numCPU** viene inizializzato a 0.

Scrivere il corpo delle seguenti funzioni C++.

1. **bool build(gestCPU* g, const char nome[])** che ripete le azioni seguenti fino ad arrivare alla fine del file: legge due stringhe di al massimo **LUN** caratteri, un numero **float** ed un numero intero che costituiscono, rispettivamente, i campi **prod**, **ident**, **costo** e **quanti** di una struttura di tipo **CPU** ed inserisce la struttura nel sistema di gestione; la funzione restituisce **false** se con i dati contenuti nel file si possono riempire un numero di strutture superiore a **DIM**; altrimenti restituisce **true**.
2. **void ordina(gestCPU* g)** che ordina il contenuto del sistema di gestione in modo tale che le CPU siano ordinate per valori alfabeticamente crescenti dei loro produttori, e quando il produttore è uguale siano ordinate per valori crescenti del costo.
3. **bool prezzo(const gestCPU* g, const char ident[], float* p, int* quanti)** che restituisce, utilizzando il numero **float** indirizzato da **p** ed il numero **int** indirizzato da **quanti**, il costo e la quantità di CPU identificate da **ident**. La funzione restituisce **true** se il tipo di CPU è presente nel sistema di gestione; altrimenti restituisce **false**.
4. **bool elimina(gestCPU* g, const char ident[])** che elimina la CPU identificata da **ident** e ricompatta il vettore **vettCPU** in modo tale da mantenere il vettore ordinato (si supponga che sia stata già eseguita la funzione **ordina**); la funzione restituisce **true** se il tipo di CPU è presente nel sistema di gestione; altrimenti restituisce **false**.
5. **void salva(const gestCPU* g, const char nome[])** che salva sul file il cui nome è passato come secondo argomento alla funzione, il produttore, l'identificatore, il costo e la quantità di tutte le CPU che costano meno di 80 Euro e la cui quantità è maggiore di 10.

NOTE SULLO SVOLGIMENTO DELLA PROVA:

- PER SVOLGERE L'ELABORATO, 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 LO SVOLGIMENTO DELL'ELABORATO, selezionare la voce **Identifica studente** nel menù **Strumenti** all'interno dell'ambiente Dev-C++ e inserire i dati richiesti;
- Per svolgere l'elaborato, 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);
- SE L'ELABORATO È STATO COMPILATO SENZA ERRORI, PRIMA DELLA CONSEGNA, selezionare la voce **Consegna Compito** nel menù **Strumenti** all'interno dell'ambiente Dev-C++ e premere il tasto INVIO fino a quando non sparisce la finestra che è stata attivata.