

Un sistema per la gestione di richieste di case popolari può essere rappresentato come una coda attraverso le seguenti costanti e strutture:

```
const int MAXNUM = 20;
const int MAXL = 50;
const float SOGLIA = 5000.0;
```

```
struct Richiesta
{ int nc;
  int va;
  float reddito;
  char nome[MAXL];
  char cognome[MAXL];
};

struct CodaRichieste
{ Richiesta ric[MAXNUM];
  int front, back;
};
```

La struttura **CodaRichieste** rappresenta una coda di richieste ed è realizzata mediante un array di dimensione **MAXNUM** (quindi il numero massimo di richieste che possono essere memorizzate è pari a **MAXNUM-1**). **front** e **back** sono gli indici utilizzati rispettivamente per le operazioni di estrazione e inserimento nella coda. Quando viene creata una istanza di **CodaRichieste** a **front** e a **back** viene assegnato il valore 0. Ogni richiesta è caratterizzata da: un intero **nc** che indica il numero di componenti del nucleo familiare, un campo **va** che rappresenta il numero di vani necessari alla famiglia, un reale **reddito** che indica il reddito complessivo della famiglia, due stringhe (di lunghezza **MAXL**) **nome** e **cognome** che identificano il richiedente. Si definisce inoltre reddito pro-capite il reddito di una famiglia diviso per il numero di componenti del nucleo familiare.

Scrivere il corpo delle seguenti funzioni C++.

1. **bool inserisci(const Richiesta r, CodaRichieste* cr)** che, se il reddito pro-capite relativo alla richiesta **r** è inferiore o uguale alla soglia, inserisce la richiesta **r** in fondo alla coda e restituisce *true*; se il reddito pro-capite della richiesta **r** è maggiore della soglia (**SOGLIA**) o se non c'è spazio in coda non inserisce la richiesta in coda e restituisce *false*.
2. **bool nuovaUnita(int v, CodaRichieste* cr, Richiesta* pr)** che rimuove dalla coda la richiesta che necessita di un numero di vani esattamente pari a **v** presente da più tempo nella coda (restituendo il valore *true*); la richiesta estratta viene restituita al chiamante attraverso l'oggetto puntato da **pr**; se nessuna richiesta viene rimossa dalla coda (perché non c'è nessuna richiesta per esattamente **v** vani o perché la coda è vuota) viene restituito il valore *false*.

3. **int posizione(const char c[], const char n[], const CodaRichieste* cr)** che restituisce la posizione in graduatoria del richiedente con nome **n** e cognome **c** (se la coda è vuota o se il richiedente specificato non è presente viene restituito il valore 0).
4. **bool salva(const CodaRichieste* cr, const char nf[])** che salva i dati relativi alle richieste presenti nella coda nel file con nome specificato da **nf**. I dati di ogni richiesta vengono salvati su una riga separata secondo il seguente formato: *cognome, nome, reddito, numero componenti, vani*. La prima riga del file corrisponde alla prima richiesta inserita in coda, la seconda riga alla seconda richiesta inserita in coda e così via. La funzione restituisce il valore *true* se l'operazione va a buon fine, *false* in caso di errore.
5. **int perreddito(const CodaRichieste* cr, Richiesta ar[])** che riempie l'array **ar** con le richieste presenti in coda in modo tale che siano ordinate secondo un reddito pro-capite crescente. La funzione restituisce il numero di elementi inseriti nell'array **ar**.

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.