

**NOTE SULLO SVOLGIMENTO DELLA PROVA:**

- Fare partire il Dev-C++ (dal Menù **Avvio** (o **Start**) nella barra degli strumenti in fondo allo schermo, selezionare Programmi e quindi Dev-C++);
- attraverso il menu **File->Open...** (o **apri** se il Dev-C++ è in italiano) aprire il progetto *APRIQUESTO.dev* presente nella cartella *c:\esame\esaInf*;
- scrivere le funzioni richieste nel file *compito.cpp* già presente nel progetto;
- salvare spesso in modo da non perdere il lavoro nel caso in cui il PC abbia un malfunzionamento (per salvare premere il bottone a forma di dischetto del Dev-C++);
- 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).

**Voli**

Un programma per la gestione di voli può essere definito a partire dalle seguenti costanti e dai seguenti tipi:

```
const int MAXS = 20;
const int RIG = 10;
const int COL = 4;
const int MAXV = 30;

struct Volo {
    int codice;
    char partenza[MAXS];
    char destinazione[MAXS];
    bool posti[RIG][COL];
};

struct ArchivioVoli{
    Volo vol[MAXV];
    int numVoli;
};
```

`Volo` memorizza il codice univoco del volo, l'aeroporto di partenza e quello di destinazione. Inoltre, il campo `posti`, indica quali posti sul volo sono ancora liberi (valore *true*) e quali sono occupati (valore *false*). I posti sono organizzati in `RIG` righe e `COL` colonne. `ArchivioVoli` memorizza i dati di un insieme di voli. Il numero di voli è indicato dal campo `numVoli` mentre il campo `vol` contiene i dati veri e propri.

Realizzare le seguenti funzioni:

**bool carica(ArchivioVoli\* pa, const char nomf[])** carica dal file indicato dall'argomento `nomf` i dati di un insieme di voli. Il file contiene un numero imprecisato di righe e ogni riga contiene nell'ordine: il codice del volo (un intero), l'aeroporto di partenza (una parola), l'aeroporto di destinazione (una parola). Tutti i posti di ogni volo sono inizialmente liberi. La funzione restituisce *true* se il caricamento va a buon fine, *false* altrimenti.

**bool prenota(ArchivioVoli\* pa, int c, int\* pr, int\* pc)** prenota, se disponibile, un posto sul volo con codice `c`. La riga e la colonna del posto prenotato vengono restituiti al chiamante mediante `pr` e `pc`. La funzione restituisce *true* se la prenotazione ha successo, *false* altrimenti.

**void stampaVolo(const ArchivioVoli\* pa, int c)** stampa a video le informazioni relative al volo con codice `c`. Rispettare il formato illustrato dal seguente esempio (L -> posto libero, O -> posto occupato):

```
3456   Roma   Parigi
OOOL
LLLL
```

```
LLLL
LLLL
LLLL
LLLL
LLLL
LLLL
LLLL
LLLL
```

**void stampaFrazioni(const ArchivioVoli\* pa)** stampa a video, per ogni volo, la frazione di posti occupati. Esempio:

```
3456 0.075
2121 0.2
9876 0
8765 0
6666 0.2
1818 0
2323 0.05
4341 0
7707 0
```

**int rimuovi(ArchivioVoli\* pa, const char p[], const char d[])** rimuove tutti i voli in cui il nome dell'aeroporto di partenza contiene la sottostringa `p` e il nome di quello di arrivo contiene la sottostringa `d`. La funzione restituisce il numero di voli rimossi.