

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).

Turni

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

```
const int MAXS = 50;
const int MAXO = 100;
const int ORE = 8;
const int GIORNI = 7;
```

```
enum Turno {RIPOSO, POSTAZIONE_A, POSTAZIONE_B, POSTAZIONE_C};
```

```
struct Orario {
    char lavoratore[MAXS];
    Turno tu[ORE][GIORNI];
};
```

```
struct ArchivioOr {
    Orario orar[MAXO];
    int numOrar;
};
```

Orario contiene il programma di lavoro settimanale di un lavoratore. Il nome del lavoratore è contenuto nel campo *lavoratore*. Il campo *tu* è una tabella con un numero di colonne pari al numero di giorni della settimana e tante righe quante sono le ore della giornata. Le singole caselle della tabella indicano se nel giorno e ora in questione il lavoratore è a riposo o è al lavoro presso una delle possibili postazioni (come indicato dal tipo enumerazione *Turno*). *ArchivioOr* contiene gli orari di lavoro di un certo numero di lavoratori (nel campo *orar*). Il numero effettivo di orari è indicato dal campo *numOrar*.

Realizzare le seguenti funzioni:

void stampa(const ArchivioOr* pa, const char n[]) stampa a video l'orario settimanale del lavoratore avente nome *n*. La stampa è costituita da un numero di colonne pari al numero di giorni della settimana e un numero di righe pari al numero di ore. Ogni ora è rappresentata da uno dei seguenti caratteri: R per riposo, A per postazione A, B per postazione B, C per postazione C. Esempio:

```
Nome? Pino
ARRRRRR
BRRRRRR
CRRRRRR
ARRRRRR
BRRRRRR
CRRRRAR
RRRRRRR
RRRRRRR
```

bool scambia(ArchivioOr* pa, const char n1[], const char n2[], int g) scambia le attività del lavoratore con nome *n1* con quelle del lavoratore con nome *n2* relativamente al giorno indicato da *g*. La funzione restituisce *true* se lo scambio va a buon fine, *false* altrimenti (uno o entrambi lavoratori non presenti in archivio).

void stampaStatistiche(const ArchivioOr* pa) stampa a video le statistiche di tutti gli orari presenti nell'archivio. In particolare per ogni orario deve essere stampato: il nome del lavoratore, la frazione di ore di riposo, la frazione di ore in postazione A, B e C (rispetto al totale delle ore settimanali). Esempio:

Mario R: 0.910714 A: 0.0535714 B: 0.0178571 C: 0.0178571

Gino R: 0.892857 A: 0.0357143 B: 0 C: 0.0714286

Pino R: 0.875 A: 0.0535714 B: 0.0357143 C: 0.0357143

bool conPiuOre(const ArchivioOr* pa, char n[]) trova il lavoratore che fa più ore di lavoro (non di riposo) tra quelli presenti nell'archivio. Il nome del lavoratore viene restituito al chiamante mediante il parametro *n*. La funzione restituisce *true* se l'operazione va a buon fine (c'è almeno un orario nell'archivio), *false* altrimenti (archivio vuoto).

bool carica(ArchivioOr* pa, const char fn[]) riempie l'archivio con i dati caricati dal file di nome *fn*. Il file ha il formato illustrato dal seguente esempio:

```
Mario 5
1 1 A
1 5 A
2 3 B
2 4 C
4 1 A
Gino 6
1 2 C
3 3 C
4 3 C
5 3 C
0 4 A
1 4 A
Pino 7
0 0 A
1 0 B
2 0 C
3 0 A
4 0 B
5 0 C
5 5 A
```

Il file contiene un numero non noto a priori di orari. Per ogni orario il file contiene: una parola che indica il nome del lavoratore a cui tale orario si riferisce, un intero che indica il numero di ore non di riposo (chiamiamo *q* tale valore). Seguono *q* righe ognuna delle quali contiene: indice dell'ora di lavoro (un intero), indice del giorno di lavoro (un intero), postazione di lavoro (carattere A, B, o C). Tutte le altre ore sono di riposo. Lo stesso schema si ripete per gli altri orari presenti nel file. La funzione restituisce *true* se il caricamento va a buon fine, *false* altrimenti.