

**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 nel directory *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).

**Attività**

L'esecuzione di attività può essere rappresentata a partire dalle seguenti costanti e dai seguenti tipi:

```
const int TICK = 20;
const int MAXU = 100;
const int DIM = 50;
const int MAXS = 20;
```

```
struct Attivita {
    bool prio;
    int durata;
    int utente;
};
```

```
struct Utente {
    int id;
    char nome[MAXS];
};
```

```
struct Coda {
    Attivita att[DIM];
    int front;
    int back;
};
```

```
struct ListaU{
    Utente ut[MAXU];
    int numUt;
};
```

Una *Attivita* è caratterizzata da una priorità (se il booleano *prio* vale *true* la priorità è alta, altrimenti è bassa), una durata residua (espressa in unità di tempo), e da un campo che contiene l'identificatore dell'utente proprietario di quella attività. Il tipo *Utente* memorizza l'identificatore univoco di un utente (un intero) e il suo nome (una stringa). Le attività sono memorizzate in una *Coda*, mentre gli utenti sono memorizzati in una *ListaU* (*numUt* indica il numero di utenti presenti).

Scrivere il corpo delle seguenti funzioni C++.

- **int esegui(Coda\* pc, int qt)** porta avanti l'esecuzione delle attività secondo la seguente logica: l'attività che si trova in testa alla coda viene eseguita per un tempo pari a un tick (la cui durata in unità di tempo è specificata dall'omonima costante) e la sua durata residua viene diminuita in modo corrispondente; se l'attività in

questione è arrivata a conclusione (durata residua pari a zero) viene rimossa dalla coda, altrimenti viene rimessa in fondo alla coda in attesa di essere eseguita nuovamente. L'esecuzione di attività procede per un numero di tick pari a quanto specificato dal parametro *qt*. La funzione restituisce il numero di attività portate a conclusione<sup>1</sup>.

- **void stampaStato(const Coda\* pc, const ListaU\* pl)** stampa lo stato della coda e della lista di utenti. Il formato deve essere analogo al seguente:

```
-----  
<100, 1234!>  
<200, 1234>  
<900, 3333!>  
<20, 9876>  
<500, 3333!>  
<400, 3333!>  
-----  
Mario, 1234  
Pino, 3333  
Furio, 9876  
-----
```

Per le attività vengono riportati, nell'ordine, durata residua e id dell'utente. Il punto esclamativo denota le attività ad alta priorità. Per gli utenti vengono visualizzati il nome e l'id.

- **int rimuovi(Coda\* pc, const ListaU\* pl, const char n[])** rimuove dalla coda tutte le attività associate all'utente avente il nome specificato dal parametro *n* (si supponga che non vi siano casi di omonimia). La funzione restituisce il numero di attività rimosse.
- **int durataCumulativa(const Coda\* pc, const int ids[], int n)** la funzione riceve in ingresso un insieme (*ids*) di identificatori di utenti (la dimensione dell'insieme è pari ad *n*) e restituisce la durata complessiva di tutte le attività di tutti gli utenti specificati.
- **bool salvaOrd(const Coda\* pc, const ListaU\* pl, const char nf[])** salva nel file di nome *nf* gli identificatori di tutti gli utenti associandoli al numero di attività presenti in coda relative a tale utente. Le coppie cosiffatte devono essere ordinate per valori decrescenti del numero di attività. Seguire il formato illustrato dal seguente esempio:

```
3333, 3  
1234, 2  
9876, 1
```

La funzione restituisce *true* se il salvataggio va a buon fine, *false* altrimenti.

---

<sup>1</sup> Esempio: supponiamo che nella coda siano presenti tre attività, che chiameremo A1, A2 e A3, e che queste abbiano una durata residua rispettivamente pari a 1000, 300, e 400 unità di tempo. Se la funzione eseguita viene chiamata con un valore di *qt* uguale a due

- l'attività in testa alla coda (A1) viene eseguita per un tempo pari a un tick, la sua durata residua diviene pari a 980 e viene rimessa in fondo alla coda;
- l'attività A2, adesso in testa alla coda, viene eseguita per un tempo pari a un tick, la sua durata residua diviene pari a 280 e viene rimessa in fondo alla coda.

Alla fine la coda contiene nell'ordine le attività A3, A1, A2, con durate residue rispettivamente pari a 400, 980, 280.