

## INGEGNERIA ELETTRONICA

Un sistema di gestione, molto semplificato, dei conto correnti di una banca può essere rappresentato in C++ dalle strutture seguenti:

```
const int N = 4; const int C = 4; const int LUNG = 20;
struct operazione
{
    int importo; int data;};
struct conto
{
    char nome [LUNG];
    operazione oper [N];
    int numOp;
    float totale;
};
struct gestcont
{
    conto conti[C]; int primo;};
```

con **LUNG** il numero massimo di caratteri che compongono il nome del proprietario del conto corrente, **N** il numero massimo di operazioni (versamenti o prelievi) effettuabili in un anno sul conto corrente e **C** il numero massimo di conti correnti. I campi **importo** e **data** della struttura **operazione** indicano l'importo prelevato o versato sul conto corrente, e la data dell'operazione (la data è espressa come numero di giorni trascorsi dall'inizio dell'anno e l'anno si considera composto da 365 giorni). I campi **numOp** e **totale** della struttura **conto** indicano il numero di operazioni effettuate durante un anno ed il denaro totale contenuto nel conto corrente. Il campo **primo** della struttura **gestcont** individua l'indice del primo elemento vuoto del vettore **conti**. Quando un oggetto di tipo **gestcont** viene creato, il membro **primo** viene inizializzato a 0.

Scrivere il corpo delle seguenti funzioni C++.

1. **int apri(gestcont\* g, const char nome[])** che apre un conto corrente al cliente il cui nome è passato come parametro (si faccia attenzione ad inizializzare correttamente i campi **numOp** e **totale** della struttura **conto** del conto corrente appena aperto). La funzione restituisce l'indice dell'elemento dell'array **conti** riservato a contenere il conto corrente; questo indice individua univocamente il conto in ogni operazione.
2. **bool oper(gestcont\* g, int numConto, int importo, int data)** che effettua in data **data** l'operazione di importo **importo** sul conto corrente numero **numConto**. La funzione restituisce **false** se il numero del conto corrente non è valido; altrimenti restituisce **true** (si faccia attenzione ad aggiornare il campo **totale**).
3. **void stampaOr(gestcont\* g)** che stampa su uscita standard il nome, il totale ed il numero di operazioni di ogni conto corrente in modo tale che i nomi appaiano in ordine alfabetico. Si utilizzi la funzione **void stampaConto(const conto\* c)** che stampa un conto su uscita standard e che è già definita nel file **main.cpp**.
4. **void interessi(gestcont\* g)** che, a fine anno, calcola per ogni conto corrente gli interessi maturati e li somma al totale. Si consideri un tasso di interesse pari al 3% annuo.

Si ricorda che:

- il compito deve essere svolto aprendo il progetto *esaInf.dev* nel directory *c:\esame\esaInf* e scrivendo le funzioni richieste nel file *compito.cpp*;
- per una corretta stampa dell'elaborato bisogna mantenere il codice entro i margini imposti dall'ambiente Dev-C++.
- per la produzione dei risultati, aprire una finestra DOS, digitare **cd c:\esame\esaInf** e premere il tasto INVIO; quindi digitare **esaInf.exe > output.txt** e premere due volte il tasto INVIO.