



### Esercizio 3 (9 punti)

Si definisca una struttura `banca` che implementa una banca con massimo 10 correntisti, identificati per nome. Si scrivano i file `banca.h` e `banca.c` implementando le funzioni:

- `struct banca *construct_banca()` che restituisce una nuova banca, ancora senza correntisti;
- `void destruct_banca(struct banca *this)` che dealloca la banca `this`;
- `void deposita(struct banca *this, char *nome, double soldi)`, che aggiunge i soldi indicati sul conto del correntista chiamato `nome` che si trova nella banca `this`. Se il correntista non è ancora presente nella banca e non si è ancora arrivati al massimo di 10 correntisti, esso viene aggiunto come nuovo correntista con deposito iniziale pari a `soldi`. Se il correntista non è ancora presente nella banca e si è già arrivati a 10 correntisti, questa funzione non fa nulla;
- `void interessi(struct banca *this, double percent)`, che aggiungi a tutti i correntisti un interesse pari al `percent` per cento;
- `char *toString(struct banca *this)`, che restituisce una nuova stringa del tipo

```
Fausto: 113.840000
Samantha: 1023.030000
Alessandra: 11.340000
```

in cui cioè sono riportati i nomi dei correntisti e l'entità del deposito che hanno presso la banca.

Se tutto è corretto, l'esecuzione del seguente programma:

```
#include <stdlib.h>
#include <stdio.h>
#include "banca.h"

int main(void) {
    struct banca *b = construct_banca();
    char *s;
    deposita(b, "Fausto", 112.14); deposita(b, "Samantha", 1023.03);
    deposita(b, "Fausto", 13.00); deposita(b, "Alessandra", 11.34);
    printf("%s\n", s = toString(b));
    free(s);
    interesse(b, 2.50);
    printf("%s\n", s = toString(b)); free(s);
    destruct_banca(b);
    return 0;
}
```

dovrà stampare:

```
Fausto: 125.140000
Samantha: 1023.030000
Alessandra: 11.340000
```

```
Fausto: 128.268500
Samantha: 1048.605750
Alessandra: 11.623500
```