

COGNOME:

NOME:

MATRICOLA:

Esame di Programmazione I, 3 settembre 2012. 2 ore

Esercizio 1 [13 punti] Si scriva una funzione

```
char *binario(int n)
```

che restituisce una nuova stringa che contiene la rappresentazione binaria del numero n . Si assuma che sia $n > 0$.

Tale programma deve avere anche una funzione `main` che:

1. legge da tastiera un numero n maggiore di 0; se non è maggiore di 0 lo richiede ad oltranza;
2. chiama `binario` su tale numero e ne stampa la stringa risultante;
3. dealloca la stringa risultante.

Se tutto è corretto, un'esecuzione di tale programma dovrebbe essere del tipo:

```
Inserisci un numero positivo: -3
Inserisci un numero positivo: 0
Inserisci un numero positivo: 268
100001100
```

Esercizio 2 [9 punti] Si considerino le liste viste a lezione. Si implementi una funzione ricorsiva:

```
struct list *put0(struct list *this);
```

che restituisce una lista ottenuta da `this` inserendo, davanti agli elementi di valore dispari, un nuovo nodo che contiene il valore 0.

Se tutto è corretto, l'esecuzione del programma:

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

int main(void) {
    struct list *l =
        construct_list
        (11, construct_list
        (13, construct_list
        (-4, construct_list
        (13, construct_list
        (-5, construct_list
        (0, construct_list
        (1, NULL))))));

    print_list(l);
    printf("\n");

    print_list(put0(l));
    printf("\n");

    return 0;
}
```

dovrà stampare:

```
[11, 13, -4, 13, -5, 0, 1]
[0, 11, 0, 13, -4, 0, 13, 0, -5, 0, 0, 1]
```

Esercizio 3 [10 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, al momento 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 preleva(struct banca *this, char *nome, double soldi)`, che preleva i `soldi` indicati dal conto del correntista `nome`, riducendo quindi i soldi che `nome` ha depositati sul conto;
- `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);

    preleva(b, "Fausto", 11.30);

    printf("%s\n", s = toString(b));
    free(s);

    destruct_banca(b);
    return 0;
}

```

deve stampare:

```

Fausto: 125.140000
Samantha: 1023.030000
Alessandra: 11.340000

Fausto: 113.840000
Samantha: 1023.030000
Alessandra: 11.340000

```