## Esame di Programmazione I

2 febbraio 2015 (tempo disponibile: 2 ore)

```
Esercizio 1 (11 punti)
```

Si scriva una funzione char \*histo(int values[], int length) che restituisce una nuova stringa che rappresenta una barra orizzontale di 80 caratteri divisa in segmenti di lunghezza proporzionale ai valori dell'array values. Il paramentro length è la lunghezza di tale array. I segmenti sono fatti, alternativamente, dai caratteri \*@+#. Se tutto è corretto, l'esecuzione del seguente programma:

Esercizio 2 (11 punti)

Si considerino le liste di char come viste a lezione. Si definisca una funzione ricorsiva

```
int *frequency(struct list *this)
```

che restituisce un nuovo array di interi contenente le frequenze di ciascun carattere nella lista this. Questo significa che il primo elemento dell'array sarà il numero di caratteri 'a' che si trovano in this, il secondo elemento sarà il numero di caratteri 'b' che si trovano in this e così via. Per esempio, l'esecuzione del seguente programma:

```
int main(void) {
  struct list *l = construct_list
    ('f', construct_list('a', construct_list('f', construct_list ('z', construct_list
        ('a', construct_list('f', construct_list('k', construct_list('m', NULL)))))));
  printf("l = "); print_list(l); printf("\n");
  int *freq = frequency(1);
  int pos;
 for (pos = 0; pos < 26; pos++)
   printf("%i ", freq[pos]);
 printf("\n");
 free(freq);
  return 0;
}
dovrà stampare
l = [f, a, f, z, a, f, k, m]
2000030000101000000000000000
```

## 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.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:

Alessandra: 11.623500

```
#include <stdlib.h>
#include <stdio.h>
#include "banca.h"
int main(void) {
  struct banca *b = construct_banca();
  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
```