

Esame di Programmazione I

15 settembre 2014 (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 `length` istogrammi orizzontali fatti da asterischi. Il primo è fatto da `values[0]` asterischi; il secondo da `values[1]` asterischi ecc.

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

```
int main(void) {
    int values[] = { 3, 8, 0, 1, 5 };
    char *s = histo(values, 5);
    printf("%s", s);
    free(s);
    return 0;
}
```

dovrà stampare:

```
***
*****

*
*****
```

Esercizio 2 (11 punti)

Si considerino le liste di `int` come viste a lezione. Si definisca una funzione **ricorsiva**

```
int compare_sum(struct list *this, struct list *that)
```

che restituisce un intero negativo se la somma degli elementi di `this` è maggiore della somma degli elementi di `that`; un intero positivo se la somma degli elementi di `that` è maggiore della somma degli elementi di `this`; e ritorna 0 altrimenti. Per semplicità, si assuma che `this` e `that` abbiano la stessa lunghezza. La funzione `compare_sum` non deve richiamare altre funzioni ricorsive, se non se stessa.

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

```
int main(void) {
    struct list *l1 = construct_list
        (2, construct_list(5, construct_list(4, construct_list(-3, NULL))));
    struct list *l2 = construct_list
        (-5, construct_list(13, construct_list(16, construct_list(26, NULL))));
    printf("The Answer to The Ultimate Question of Life: %i\n", compare_sum(l1, l2));
    return 0;
}
```

dovrà stampare un numero positivo (non necessariamente 42...):

The Answer to The Ultimate Question of Life: 42

Esercizio 3 (10 punti)

Si scrivano i file `casa.h` e `casa.c` che definiscono una casa, con le tre metrature essenziali che possono interessare un'agenzia immobiliare, con le seguenti funzioni sulle case:

```
struct casa *construct_casa(int metratura_casa, int metratura_terrazzi, int metratura_garage);
void destroy_casa(struct casa *this);
double metratura_commerciale(struct casa *this);
```

La funzione di costruzione `construct_casa` richiede di specificare i metri quadri della casa, dei suoi eventuali terrazzi e del suo eventuale garage. La metratura commerciale di una casa è definita come i metri quadri della casa, più il 70% dei metri quadri dei terrazzi più il 50% dei metri quadri del garage.

Quindi si definiscano i file `vendita.h` e `vendita.c` che definiscono una vendita immobiliare di una casa. Una vendita è specificata dalla casa e dal prezzo a cui è venduta. Devono essere definite e implementate le seguenti funzioni:

```
struct vendita *construct_vendita(struct casa* casa, int prezzo);
void destroy_vendita(struct vendita *this);
double prezzo_m2(struct vendita *this);
```

dove la funzione `prezzo_m2` calcola il prezzo della vendita al metro quadro (prezzo della vendita diviso i metri quadri commerciali della casa venduta).

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

```
#include <stdio.h>
#include "casa.h"
#include "vendita.h"

int main(void) {
    struct casa *c1 = construct_casa(80, 20, 15);
    struct casa *c2 = construct_casa(70, 0, 0);
    printf("c1 ha una metratura commerciale di %.2f m2\n", metratura_commerciale(c1));
    printf("c2 ha una metratura commerciale di %.2f m2\n", metratura_commerciale(c2));
    // la stessa casa viene venduta due volte a prezzi diversi
    struct vendita *v1 = construct_vendita(c1, 200000);
    struct vendita *v2 = construct_vendita(c1, 250000);
    printf("c1 e' stata venduta una prima volta a %.2f euro al m2\n", prezzo_m2(v1));
    printf("c1 e' stata venduta una seconda volta a %.2f euro al m2\n", prezzo_m2(v2));

    return 0;
}
```

dovrà stampare:

```
c1 ha una metratura commerciale di 101.50 m2
c2 ha una metratura commerciale di 70.00 m2
c1 e' stata venduta una prima volta a 1970.44 euro al m2
c1 e' stata venduta una seconda volta a 2463.05 euro al m2
```