

COGNOME:

NOME:

MATRICOLA:

Esame di Programmazione I, 24 settembre 2012. 2 ore

Esercizio 1 [14 punti] Si scriva una funzione

```
char *replace(char *s, char c, char *replacement)
```

che restituisce una nuova stringa ottenuta sostituendo con `replacement` ogni occorrenza di `c` dentro `s`.

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

```
int main(void) {
    char *s = "ciao amico";
    char *risultato;

    printf("%s\n", risultato = replace(s, 'a', "hello"));
    free(risultato);

    printf("%s\n", risultato = replace(s, 'u', "hello"));
    free(risultato);

    // si noti: il terminatore \0 non e' un carattere della stringa!
    printf("%s\n", risultato = replace(s, '\0', "hello"));
    free(risultato);

    return 0;
}
```

deve stampare:

```
cihelloo hellomico
ciao amico
ciao amico
```

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

```
struct list *fib(int n);
```

che restituisce la lista dei primi `n` numeri di Fibonacci, in ordine inverso.

Se tutto è corretto, l'esecuzione del programma:

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

int main(void) {
    print_list(fib(10));
    printf("\n");

    print_list(fib(0));
    printf("\n");

    return 0;
}
```

dovrà stampare:

```
[55, 34, 21, 13, 8, 5, 3, 2, 1, 1]
[]
```

Esercizio 3 [9 punti] Si definisca una struttura `studente` che implementa uno studente. Si scrivano i file `studente.h` e `studente.c` implementando le funzioni:

- `struct studente *construct_studente(char *nome)` che restituisce un nuovo studente con il nome indicato;
- `void destruct_studente(struct studente *this)` che dealloca lo studente `this`;
- `void fa_esame(struct studente *this, int voto)`, che registra il voto indicato per lo studente `this`, se il voto è fra 18 e 30 inclusi, e non fa nulla altrimenti; non c'è limite al numero di esami che uno studente può fare;
- `float media(struct studente *this)`, che restituisce la media dei voti degli esami sostenuti dallo studente `this`; se lo studente non ha ancora fatto esami, restituisce 0.0.
- `char *toString(struct studente *this)`, che restituisce una nuova stringa fatta dal nome dello studente `this` seguito dalla media degli esami sostenuti da `this`.

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

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

int main(void) {
    struct studente *s1 = construct_studente("Giacomo");
    struct studente *s2 = construct_studente("Elisa");
    char *s;

    fa_esame(s1, 18);
    fa_esame(s1, 15); // non viene registrato
    fa_esame(s2, 30);
    fa_esame(s1, 25);
    fa_esame(s2, 22);
    fa_esame(s2, 29);
    fa_esame(s2, 27);

    printf("%s\n", s = toString(s1));
    free(s);

    printf("%s\n", s = toString(s2));
    free(s);

    destruct_studente(s1);
    destruct_studente(s2);

    return 0;
}
```

deve stampare:

```
Giacomo  21.50
Elisa    27.00
```