

Ingegneria degli Algoritmi (A.A. 2012-2013)

Corsi di Laurea in Ingegneria Informatica e Automatica, Ingegneria dei Sistemi Informatici, e Laurea Magistrale in Ingegneria Informatica

Sapienza Università di Roma

Appello del 20/09/2013 – Prova al calcolatore – Durata 1h 30'

Preliminari (aula 17)

1. Fare login in Linux (Scientific Linux 5.1) con le credenziali fornite in aula
2. La directory `/local/studente/Desktop/esame` contiene:
 - a. `bintree`: directory di lavoro per l'esercizio 1;
 - b. `substring`: directory di lavoro per l'esercizio 2;
 - c. `studente`: file in cui inserire nome, cognome e matricola.

Esercizio 1: eliminazione delle foglie di un albero binario (16 punti su 32)

Si richiede di scrivere una funzione C che elimini tutte le foglie da un albero binario i cui nodi sono definiti come segue (si veda `bintree/bintree.h`):

```
typedef struct bintree bintree;  
struct bintree {  
    void* elem;  
    bintree *left, *right;  
};
```

Dove `elem` è un puntatore al contenuto informativo associato al nodo, mentre `left` e `right` sono i puntatori al figlio sinistro e destro nell'albero, rispettivamente (o `NULL` se i figli sono assenti). La funzione, da definire nel file `bintree/bintree.c`, deve avere il seguente prototipo:

```
void bintree_remove_leaves(bintree* root)
```

Dove `root` è la radice dell'albero binario.

Compilazione e test.

- *Directory di lavoro*: `bintree/`
- *Compilazione programma*: dare il comando `make`, che genera il file eseguibile `bintree`;
- *Test funzionamento*: dare il comando `make test`;
- *Test uso corretto della memoria*: dare il comando `make test-valgrind`. Verificare che `valgrind` non riporti errori di accesso a memoria e `memory leak`.

Esercizio 2: eliminazione di sottostringa (16 punti su 32)

Si richiede di scrivere una funzione C che, data una stringa `char* s` e due indici `int i` e `j`, con $i \leq j$, elimini dalla stringa tutti i caratteri con indici compresi tra `i` e `j-1` compresi. La funzione deve poter essere **invocata** come segue (si veda la funzione `check` in `bintree/main.c`):

<code>substring_remove(&s, i, j)</code>

La stringa data in ingresso deve essere riallocata in modo che il numero di byte allocati sia ridotto di `j-i` byte. Se `i` o `j` sono maggiori del numero di caratteri della stringa, vengono automaticamente considerati come uguali alla lunghezza della stringa. Se sono minori di zero, vengono automaticamente considerati come uguali a zero.

Compilazione e test.

- *Directory di lavoro:* `substring/`
- *Compilazione programma:* dare il comando `make`, che genera il file eseguibile `substring`
- *Test funzionamento:* dare il comando `make test`.
- *Test uso corretto della memoria:* dare il comando `make test-valgrind`.
Verificare che `valgrind` non riporti errori di accesso a memoria e `memory leak`.