

Parziale di Programmazione I - Bioinformatica

6 febbraio 2024, turno delle 14:30 (tempo disponibile: 2 ore)

Esercizio 1 (19 punti) (si consegna letters.c e letters.h)

Si completi il seguente file letters.c:

```
// aggiungete #include e funzioni ausiliarie se servono

// inizializza arr, lungo length, con lettere casuali dell'alfabeto
// inglese, maiuscole o minuscole, in modo tale che non ci siano
// mai due maiuscole o due minuscole di seguito; non randomizza
// la sequenza casuale con srand(time(NULL)): ci pensa il main
void init(char arr[], int length) {
    // completare
}

// stampa (senza fare spazi in mezzo) i caratteri di arr, lungo length,
// e va a capo alla fine
void print(char arr[], int length) {
    // completare
}

// modifica l'array arr, lungo length, in modo che le vocali
// vengano sostituite con la vocale successiva, in modo ciclico
// (cioe' la 'a' diventa una 'e', la 'e' diventa una 'i' ecc;
// la 'u' diventa una 'a'; stesse modifiche per le vocali maiuscole)
void rotate_vowels(char arr[], int length) {
    // completare
}
```

Si scriva quindi un file di header letters.h che dichiara le precedenti tre funzioni.

Per esempio, il seguente file main.c (già fornito e da non modificare):

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "letters.h"

int main(void) {
    srand(time(NULL)); // inizializza la sequenza casuale
    char array[70];    // crea l'array
    init(array, 70);   // inizializza l'array in modo casuale
    printf("array prima: ");
    print(array, 70);  // stampa l'array
    rotate_vowels(array, 70); // ruota le vocali
    printf(" array dopo: ");
    print(array, 70);  // stampa nuovamente l'array
    return 0;
}
```

deve stampare qualcosa del tipo:

```
array prima: bCkKiWpKsSuIrHrAaTlLfZtSqWoLkYpToXoSpJiRmUdTpAtKqBwXuRcJsEtJsZxDzBmKjF
array dopo:  bCkKoWpKsSaOrHrEeTlLfZtSqWuLkYpTuXuSpJoRmAdTpEtKqBwXaRcJsItJsZxDzBmKjF
```

Esercizio 2 (12 punti) (si consegnì triangle.c)

Si completi il seguente programma `triangle.c`. La funzione `triangle` deve essere ricorsiva:

```
#include <stdio.h>

// stampa un triangolo rettangolo di altezza h, con
// l'angolo retto in alto a sinistra, con bordi sinistro e destro
// fatti dal carattere chiocciolina e con l'interno riempito dalle
// lettere alfabetiche minuscole decrescenti, fino ad arrivare alla 'a'
// in basso (si veda la stampa di esempio)
// QUESTA FUNZIONE DEVE ESSERE RICORSIVA
void triangle(int h) {
    // completare
}

int main(void) {
    triangle(5);
    triangle(8);
    return 0;
}
```

Se tutto è corretto, tale programma deve stampare due triangoli, il primo di altezza 5 e il secondo di altezza 8:

```
@ccc@
@bb@
@a@
@@
@
@ffffff@
@eeee@
@dddd@
@ccc@
@bb@
@a@
@@
@
```

Se servisse, negli esercizi si possono aggiungere funzioni ausiliarie.