14 Novembre 2016

Programmazione M-Z
Ingegneria e Scienze Informatiche - Cesena
A.A. 2016-2017

Elaborato 7

Data di sottomissione: entro la mezzanotte del 20 Novembre 2016.
Formato di sottomissione: un file compresso con nome `Elaborato7.zip`, contenente un unico file sorgente con nome `snake.c`

Specifiche:

- Sviluppare funzioni di libreria per poter gestire l'**oggetto serpente** nel gioco *snake*.
- Viene fornita l'implementazione dell'intero gioco, tranne l'implementazione della libreria `snake.c`. L'implementazione fa uso della libreria `curses`.
- I prototipi delle funzioni da implementare sono dichiarati nell'header `snake.h` e allegati alle specifiche.
- Il serpente è rappresentato come un *array di coordinate*.
- Le funzioni di libreria in `snake.c` si occupano di:
  - spostare il serpente in una direzione (up, down, right, left),
  - aggiungere una nuova testa al serpente in una direzione specificata (up, down, right, left) rispetto alla testa attuale,
  - rimuovere la coda del serpente,
  - ritornare le coordinate della testa attuale del serpente,
  - verificare che il serpente non si sia *annodato* (i.e. che il vettore non contenga ripetizioni delle stesse coordinate).
- Le funzioni di libreria prendono in input un vettore di coordinate `s[]` e la lunghezza attuale del serpente `length`:
  - Le coordinate attuali del serpente sono salvate nelle celle

`s[0], ..., s[length-1]`.

  - Le funzioni non conoscono la lunghezza effettiva dell'array `s[]`, questa è nota solo alla funzione chiamante. Le funzioni possono assumere unicamente che la lunghezza dell'array è almeno `length+1` (i.e. una sola cella libera).
    * Ogni accesso all'array con indici $i > length + 1$ può potenzialmente causare buffer overflow.
    * Ogni accesso all'array con indici $i < 0$ causa buffer overflow.

Vincoli:

- Le implementazioni devono aderire perfettamente ai prototipi e alle specifiche fornite.
- Le eventuali funzioni di utility della libreria devono essere *nascoste* all'esterno.
- Non è possibile utilizzare puntatori o la notazione specifica per i puntatori per lo sviluppo delle funzioni di libreria.

Suggerimenti:

- Un aspetto da considerare prima di passare all'implementazione è dove posizionare la testa del serpente nell'array.
- La funzione di di spostamento nella direzione `dir` produce un nuovo set di coordinate per l'oggetto serpente. Il nuovo set di coordinate è equivalente al set di coordinate ottenuto con la seguente procedura:
  1. aggiungiamo una nuova testa nella direzione `dir` rispetto alla vecchia testa,
  2. rimuoviamo la coda.
- Per verificare se il serpente è annodato è sufficiente verificare che le coordinate della testa non siano ripetute nell'array.

```c
#ifndef SNAKE_H
#define SNAKE_H

enum direction {UP, DOWN, LEFT, RIGHT};

struct snake {
  unsigned int x;
  unsigned int y;
};


/*
 * Returns the (coordinates of the) snake's head.
 */
struct snake snake_head(struct snake s[], unsigned int length);

/*
 * Returns 1 if the snake crosses himself, 0 otherwise.
 */
int snake_knotted(struct snake s[], unsigned int length);

/*
 * Moves the snake one step forward in the dir direction.
 */
void snake_move(struct snake s[], unsigned int length,
                                          enum direction dir);

/*
 * Increases the snake length.
 *
 * This is equivalent to:
 * - add a new head in the dir direction wrt the old head.
 */
void snake_increase(struct snake s[], unsigned int length,
                                          enum direction dir);

/*
 * Decreases the snake length.
 *
 * This is equivalent to:
 * - remove the tail of the snake.
 */
void snake_decrease(struct snake s[], unsigned int length);

#endif
```