UNIVERSIDAD DE MÁLAGA | uma.es

Dpto. Lenguajes y Ciencias de la Computación

Systems Programming and Concurrency Mid-term exam 4/4/2019

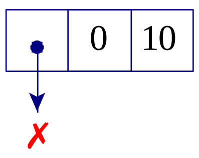
C language

We need to track into a list the trajectory of a rocket through the space. To do this, the list has to store the position and speed of the rocket in the last ten seconds. In some cases, instead of ten seconds, it should be stored more or less samples so we will use a dynamic list instead of an array.

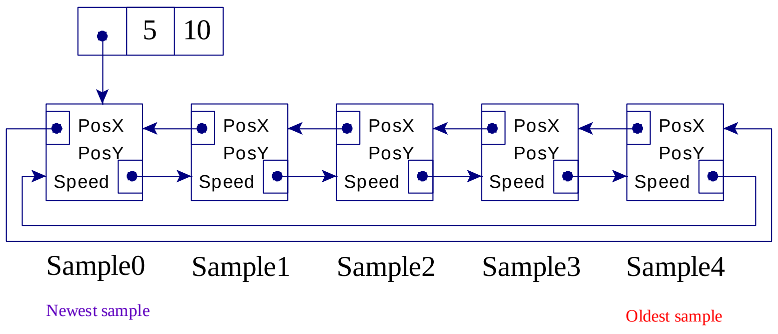
This way, the list should have ten samples at most:

- At the beginning, the list will be empty.
- As new samples are added, the list length will increase until a maximum of ten.
- Once the list is ten samples long, new samples should be written onto the oldest samples stored.

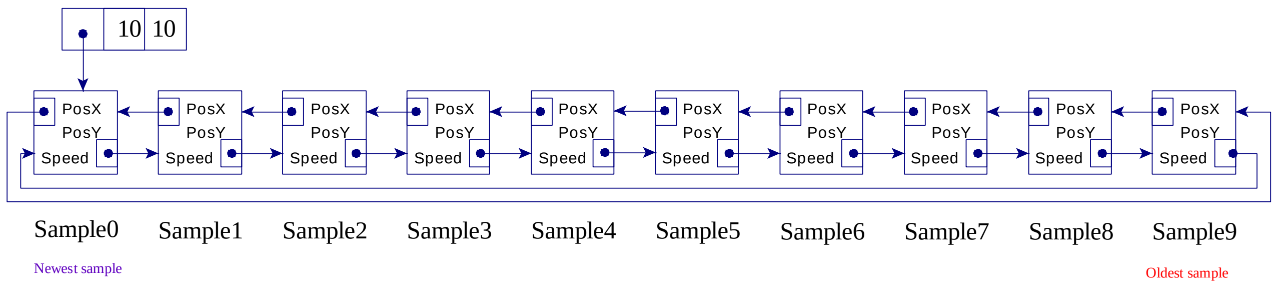
To increase performance, a circular double linked list will be used; in addition, a header is needed in order to store the current length of the list and the maximum length allowed (10). An empty list is as follows:



A list with only 5 samples is as follows:



And a fulfilled list with 10 samples is:



and cannot be enlarged anymore because it has reached to its maximum length.

To manage this list, we use the next definitions:

```c
struct info {
	int posX;
	int posY;
	int speed;
};
typedef struct T_node {
	struct info data;
	struct T_node * prev, * next;
} T_Node;
typedef struct T_list {
	struct T_node * newest;
	int length;
	int maxLength;
} T_List;
```

You have to implement the next functions given in DoubleLinkedList.h:

```c
// Creates a new empty list and returns it as a result.
T_List create();

// Inserts a new sample (x,y,s) into the list. If the list had a length lower
// than the maximum length then a new node is inserted in the head (newest).
// Otherwise, the oldest node is reused and marked as the newest one.
void insert(T_List * pl, int x, int y, int s);

// Returns the length of the list
int length(T_List l);

// Shows the content of the list, from the newest to the lowest
void show(T_List l);

// Calculates the average value of the three components of a sample. Returns the
average of posX, the average of posY and the average of speed values.
struct info avg(T_List l);

// Frees the memory used by the list and resets it (as if it were just created)
void destroy(T_List * pl);

// Saves the list in a file: first the length and maximum length, then the
samples (three pieces of data per sample: posX, posY and speed)
void save(T_List l, FILE * f);

// Loads a previously saved list. The resulting list must have the same content
and order than the original one
T_List load(FILE * f);
```

To do this, you are given a Driver.c that checks for every function.