Data Structures and Algorithms

Pierre Collet/ Manal ElZant-Karray

Practical work n°5 : Doubly/circular Linked List

Exercise 1: Write a C program to search the maximum and minimum element from a doubly linked list

To find the minimum item (maximum item) of the doubly linked list, you need to compare each item of the list by allowing a min variable (max variable) to point to the first item of the list then starting to compare its data with its next item. If the data of its next item is less than (greater than) the data of the min (max), simply allow the min (max) to catch the next item.

Exercise 2: Write a C function to check if a doubly linked list of characters is a palindrome.

a. Create a doubly linked list where each node contains only one character of a string.
b. Initialize two pointers **left** at starting of list and **right** at the end of the list.
c. Check the data at left node is equal to right node, if it is equal then increment left and decrement right till middle of the list, if at any stage it is not equal then return false.

Exercise 3: Reverse the circular Linked List implemented with SLL during the lecture session. (See the approach to reverse a singly Linked list in PW4. What do you have to add in the case of circular linked list?!)

Exercise 4 (+ Group 2): Complete the following C program to check if a Singly linked list is a palindrome.

```
#include <stdio.h>
#include <stdlib.h>
struct node{
  int num;
  struct node *next;
};
int create(struct node **);
int palin_check (struct node *, int);

int main() {
  struct node *p = NULL;
  int result, count;
  printf("Enter data into the list\n");
  count = create(&p);
  result = palin_check(p, count);
  if (result == 1)
    printf("The linked list is a palindrome.\n");
  else
    printf("The linked list is not a palindrome.\n");
  return 0;
}
```