University of Zurich

Department of Informatics
Database Technology Group
Prof. Dr. Michael Böhlen

Informatics II
Exercise 6

April 4, 2022

Pointer and Linked List

Task 1. (Pointers) Pointer is an important concept in C. This task examines knowledge about pointers in C, conceptually and practically.

a) State if the following statements about pointers are true or false:

   (a) Pointers in C language can only point to primitive data types (e.g., `int`, `double`), not non-primitive data types (e.g., `struct`, `array`).

   (b) The expressions `*ptr++` and `++*ptr` produce the same results.

   (c) If we use a primitive data type variable as the argument to a function, and modify that variable inside the function, the original data will be changed.

   (d) If we use a pointer as the argument to a function, and change the pointer inside the function, the original data will be changed.

b) Write a short C program that declares and initializes three variables: `double d`, `int i`, and `char ch`. Next declare and initialize a pointer to each of the three variables: pointer `p_d` for `d`, `p_i` for `i`, `p_ch` for `ch`. Print the following information of `d`, `i`, `ch`, `p_d`, `p_i`, and `p_ch`:

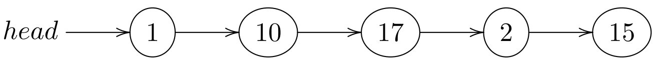
   (a) their values

   (b) their addresses

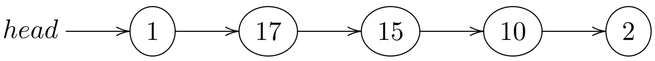
   (c) memory sizes (in bytes)

Task 2. (Grouping Linked List) There is a linked list of N nodes. Each node contains an integer value. Given the head to the linked list, create a new linked list that contains the odd values of the original linked list, followed by the even values.

For example, assume we have a linked list shown below:



After the grouping, the new linked list should be:



Tasks:

a) Implement the `insertList` function in C, which inserts a new element at the end of a list. The struct of a node in the linked list is given as below:

```
struct Node {
  int data;
  struct Node* next;
};

struct Node* head;
```

b) Implement the `displayList` function in C, which prints a given linked list.

c) Give a pseudocode of a function `groupingLinkedList`, which groups a linked list such that it contains the odd values of the original linked list, followed by the even values as described above.

d) Implement your pseudocode for grouping a linked list in C.

e) State if the following statements are true or false, justify why:

   (a) The space complexity of the function `groupingLinkedList` is $O(1)$.

   (b) The time complexity of the function `groupingLinkedList` is $O(n^2)$.

Task 3. (Anagrams Validation) An anagram is a word or phrase formed by rearranging the letters of a different word or phrase, using all the original letters exactly once. For example, `below` and `elbow` are anagrams of each other, `dusty` and `study` are anagrams of each other.

Consider each word is represented by a linked list, then the task is to validate if both linked lists are anagram to each other. This task involves three sub-tasks:

a) Write a pseudocode for this task.

b) Implement a linked list in C with `displayList` and `insertList` functions, such that each node in the list represents a letter in a word.

c) Implement the anagram's validation function in C.

Hint:

- You can assume that the two linked lists are of the same length.
- You may want to sort the linked lists.