ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет программной инженерии и компьютерной техники

Кафедра «ВТ»

ЛАБОРАТОРНАЯ РАБОТА № _4_

ПО ДИСЦИПЛИНЕ «Система языкового программирования»

Выполнил(а): Чан Чунг Дык

Группа: P3202

Санкт-Петербург

2017/2018

10.6. Assignment: Linked List

10.6.1 Assignment

The program accepts an arbitrary number of integers through stdin. What you have to do is

1. Save them all in a linked list in reverse order.

2. Write a function to compute the sum of elements in a linked list.

3. Use this function to compute the sum of elements in the saved list.

4. Write a function to output the n-th element of the list. If the list is too short, signal about it.

5. Free the memory allocated for the linked list.

11.7.2 Assignment

The input contains an arbitrary number of integers.

1. Save these integers in a linked list.

2. Transfer all functions written in previous assignment into separate .h and c files.

Do not forget to put an include guard!

3. Implement foreach; using it, output the initial list to stdout twice: the first time, separate elements with spaces, the second time output each element on the new line.

4. Implement map; using it, output the squares and the cubes of the numbers from list.

5. Implement foldl; using it, output the sum and the minimal and maximal element in the list.

6. Implement map_mut; using it, output the modules of the input numbers.

7. Implement iterate; using it, create and output the list of the powers of two (first 10 values: 1, 2, 4, 8, …).

8. Implement a function bool save(struct list* lst, const char* filename); which will write all elements of the list into a text file filename. It should return true in case the write is successful, false otherwise.

9. Implement a function bool load(struct list** lst, const char* filename);, which will read all integers from a text file filename and write the saved list into *lst. It should return true in case the write is successful, false otherwise.

10. Save the list into a text file and load it back using the two functions above. Verify that the save and load are correct.

11. Implement a function bool serialize(struct list* lst, const char* filename);, which will write all elements of the list into a binary file filename. It should return true in case the write is successful, false otherwise.

12. Implement a function bool deserialize(struct list** lst, const char* filename);, which will read all integers from a binary file filename and write the saved list into *lst. It should return true in case the write is successful, false otherwise.

13. Serialize the list into a binary file and load it back using two functions above. Verify that the serialization and deserialization are correct.

14. Free all allocated memory