Bach Khoa University
Faculty of Computer Science and Engineering

Data structure & Algorithm
Fall 2017

Lab 2 – C++ Recursion & Linked List

- C++ review
- Complexity
- Recursion

With the following struct:

```
struct node{
   int data;
   node* next = NULL;
};
```

Solve problem from 1 to 4

Problem 1

Write a function to convert a list to a linked list:

For example:

```
// Given a list with some values (data) in it
int List[5] = { 1, 2, 3, 4, 5};
int Size = 5;
// This function will retrieve all data from list
// and create a linked list from that data
// For example:
node* linkedList = ConvertToLinkedList(List, Size);//1->2->3->4->5
PrintLinkedList(linkedList); // return 1 2 3 4 5
```

Problem 2 *

Write a program having these following functions:

a) Traversal a linked list and print all data.
b) Traversal a linked list and print all data in a reserved way.
c) Sort the linked list.

For example

```
// Initialize data
int List[5] = { 5, 8, 3, 2, 9 };
int Size = 5;
node* LinkedList = ConvertToLinkedList(List, Size); // From
Problem 1

// Question 2.a - Print all data
PrintLinkedList(LinkedList); // return 5 8 3 2 9

// Question 2.b - Print all data in reserve way
PrintLinkedListReserve(LinkedList); // return 9 2 3 5 8

// Question 2.c - Sort a linked list
SortLinkedList(LinkedList, Size);
PrintLinkedList(LinkedList); // return 2 3 5 8 9
```

Problem 3*

Write a function to add (or subtract) two polymonials:

For example:

```
// Initialize data
int Poly1[5] = { 5, 8, 3, 2, 9 };
int Poly2[5] = { 2, 0, 0, 1, 5 };
node* PolyList1 = ConvertToLinkedList(Poly1, 5);
node* PolyList2 = ConvertToLinkedList(Poly2, 5);

// Question 3
node* addedPoly = AddPoly(PolyList1, PolyList2);
PrintLinkedList(addedPoly); // return 7 8 3 3 14
```

Problem 4

Write a function to add (or subtract) two polymonials:

For example:

```
// Initialize data
int Poly1[5] = { 5, 8, 3, 2, 9 };
int Poly2[5] = { 2, 0, 0, 1, 5 };
```

```
node* PolyList1 = ConvertToLinkedList(Poly1, 5);
node* PolyList2 = ConvertToLinkedList(Poly2, 5);

// Question 4
node* mulPoly = MulPoly(PolyList1, PolyList2);
PrintLinkedList(mulPoly); // return 10 16 6 9 51 43 17 19 45
```

Problem 5

Write the factorial function recursively:

Problem 6 *

Write a recursive method **sumDownBy2** with the following specification:

```
  int sumDownBy2(int n)
```

The function compute and return the sum of the positive integers n + (n-2) + (n-4) + ... +

```
sumDownBy2(7) // is 7+5+3+1 = 16
sumDownBy2(8) // is 8+6+4+2 = 20
sumDownBy2(0) // is 0
sumDownBy2(-1) // is 0
```

Problem 7 *

Given an N x N matrix composed only by 0, -1, 1, 2. Here is a 3 x 3 example:

$$\begin{matrix} 0 & -1 & 1 \\ 1 & 1 & 1 \\ 1 & -1 & 2 \end{matrix}$$

You start at 0 (always at the top left corner) and find a way to reach 2 (always at the bottom right corner). You can only travel through tiles labeled 1. When you step on a -1 tile, you will have to backtrack to the previous tile.

Write a recursive function to print the shortest part from 0 to 2 (if there is any). Hint: you can use extra functions to accomplish this task.