Tut 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 recursive function to find the max value of a linked list using `node`:

```
int myMaxFunc(node* head, int maxVal) {
      //YOUR CODE HERE
}
```

After completing the function above, draw its call stack for the following list: 13, 5, 2,

3, 32, 24, 7.

Problem 2

Given an array of integers arr with length n and the following function:

```
bool isPrime(int num) {
      if (num < 2) {
            return false;
      }

      int i;
      for (i = 2; i*i <= num; i++) {
            if (num % i == 0) return false;
      }

      return true;
}
```

Complete the below tasks:

a) Write a recursive function that print out at most one prime number in the array.

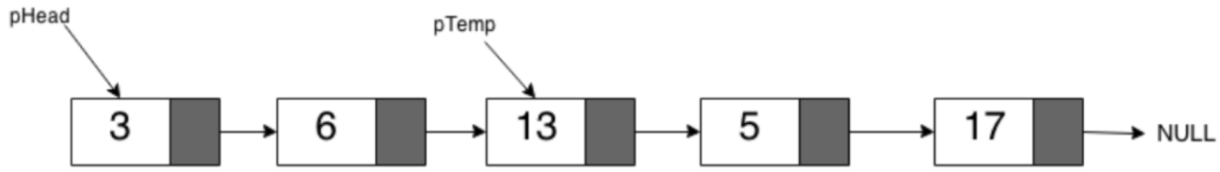
```
void onePrime(int * arr, int n) {
	//YOUR CODE HERE
}
```

b) Write a recursive function that print out all prime numbers in the array.

```
void allPrime (int * arr, int n) {
	//YOUR CODE HERE
}
```

Problem 3

Using struct `node` in **Problem 1**, suppose that we have a linked list as shown in the following figure:



Draw the linked list in which case:

a) Insert a node (value of data: 9) at the beginning of linked list.
b) Insert a node (value of data: 10) at the end of linked list.
c) Insert a node (value of data: 15) at the pTemp.
d) Delete the node which have value of data 3.
e) Delete the node which have value of data 17.
f) Delete the node which pTemp pointed.

What is the output of the following code?

g)

```
void fun1(node* head) {
	if (head == NULL)
		return;
```

```
      fun1(head->next);
      printf("%d ", head->data);
}
```

h)

```
void fun2(node* head) {
      if (head == NULL)
            return;
      printf("%d ", head->data);

      if (head->next != NULL)
            fun2(head->next->next);
      printf("%d ", head->data);
}
```

Problem 4

a) Write a function to print out all values of a linked list.

b) Suppose we have a function:

```
void func1(node* head) {
      node* temp = head;
      while (temp != NULL) {
            if (temp->next == NULL) {
                  temp->next = head;
                  return;
            }
            temp = temp->next;
      }
}
```

1. What will happen to a linked list if we pass its head pointer to the function above?
2. What is the output of the function you wrote for question a) now.
3. Propose a way to traverse along the modified list correctly.

Problem 5

Write a recursive function to return nth Fibonacci number where n given as parameter.

For example:

```
int Fibo5 = Fibonacci(5); // Fibo5 = 3
int Fibo7 = Fibonacci(7); // Fibo7 = 8
```

where the Fibonnacci sequence start with 0, 1.

a) Use your program to compute the 10th, 20th,30th and 40th Fibonacci numbers.
b) Why does it take so much longer to computer the higher Fibonacci numbers?

Problem 6

Using struct `node` in **Problem 1**, write function to search a node of a single linked list.

```
node* searchList(node* pHead, int data)
```