

Tut 2 – Recursion & Linked List

Question 1:

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
}
```

Question 2:

With the following struct:

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

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: -11, 5, 9, 3, 100, 2, 7.

Question 3:

Given a positive integer N, try to print out a block of text similar to the below pictures:

```

_ _ _ _ _ 9
_ _ _ _ _ 89
_ _ _ _ _ 789
_ _ _ _ _ 6789
_ _ _ _ _ 56789
_ _ _ _ _ 6789
_ _ _ _ _ 789
_ _ _ _ _ 89
_ _ _ _ _ 9

```

N = 9

```

_ _ _ _ _ 6
_ _ _ _ _ 56
_ _ _ _ _ 456
_ _ _ _ _ 456
_ _ _ _ _ 56
_ _ _ _ _ 6

```

N = 6

- a) Use recursion with the following prototype:

```
void recursiveTriangle(int N)
```

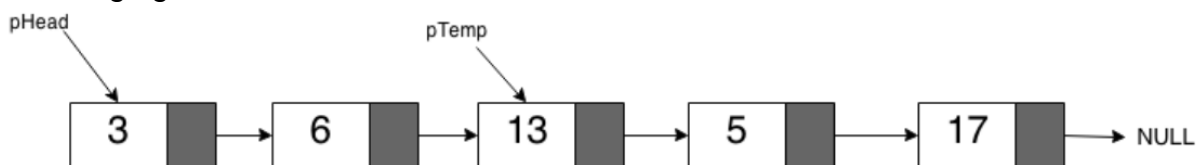
Hint: you can use extra functions to accomplish this task. They should not contain **for** loops.

- b) Use **for** loops to solve it with the following prototype:

```
void loopTriangle(int N)
```

Question 4:

Using struct `node` in **Question 2**, suppose that we have a linked list as shown in the following figure:



Draw the linked list in which case:

- Insert a node (value of data: 9) at the beginning of linked list.
- Insert a node (value of data: 10) at the end of linked list.
- Insert a node (value of data: 15) at the pTemp.
- Delete the node which have value of data 3.
- Delete the node which have value of data 17.
- 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);
}
```

Question 5:

- 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;
    }
}
```

What will happen to a linked list if we pass its head pointer to the function above?

What is the output of the function you wrote for question a) now?

Propose a way to traverse along the modified list correctly.

Question 6:

Write a function that delete the n-th node of a linked list and return the new head if n is 0 (in case we delete the first node):

```
node* deleteNth(node* head, int n)
```

If n is greater than the length of the list or less than 0, no node will be deleted.

Given a list called list1 with the following values: 4, -1, 10, 5. Illustrate step by step what would happen if you call deleteNth(list1, 3).