COMP 1603 – Tutorial/Lab 11

1. Write a recursive function to find the number of times the digit 9 occurs in a positive integer. Eg 9 occurs 2 times in 19932.

2. Write a recursive function `sumGt` to determine if all the elements in a linked list of positive integers greater than 0 have a sum greater than 10 with the function prototype: `bool sumGt(Node * top, int sum);`, where `top` is the head of the linked list and `sum` is the sum of all the nodes to check if they are greater than a given sum.

3. Write a recursive function `increment` to add 1 to all the elements of a linked list of positive integers.

4. Write a function `mergeDesc` to merge 2 linked lists of characters containing letters only in descending order which may have both upper and lowercase letters.

5. What is the output of the following:

```
#include <cstdlib>
#include <iostream>
using namespace std;

int main() {

	int x, y;
	int *px, *py, *z;
	px = &x;
	py = &y;
	x = 30;
	y = 70;
	z = &x;
	*z = y - *z;
	cout << " 1. *z is " << *z << endl;
	*py = *py + 10;
	cout << " 2. The value of *py is " << *py << endl;
	int arr[] = {12,14,16,18};
	int *ptr = arr;
	for (int j=1; j <= 3; j++) {
		(*ptr) *= 2;
		ptr++;
	}
	cout << " 3." << endl;
	for (int j=0; j < 4; j++)
		cout << arr[j] << '\t';

	return 0;
}
```

6. Write a recursive function to count the number of occurrences of a given key in an integer array with the function prototype: `int numOccur(int A[], int n, int key);`, where `A[]` is the array, `n` is the size of the array and `key` is the element to be searched.