CMPT125, Fall 2018

Final Exam
December 5, 2018

Name___________________________

SFU ID: |__|__|__|__|__|__|__|__|__|

Problem 1	
Problem 2	
Problem 3	
Problem 4	
Problem 5	
TOTAL	

Instructions:

1. Write your name and SFU ID **clearly**
2. This is a closed book exam, no calculators, cell phones, or any other material.
3. The exam contains five (5) problems.
4. Each problem is worth 20 points.
5. Write your answers in the provided space.
6. There is an extra page in the end of the exam. You may use it if needed.
7. Explain all your answers.

Good luck!

Problem 1 [20 points]

a) Consider the following function.

```
int fun(int x, unsigned int y) {
    if (y == 0)
        return 0;
    else {
        int tmp =  fun(x,y/2);
        if (y%2 == 0) // returns true if y is divisible by 2
           return tmp + tmp;
        else
           return tmp + tmp + x;
    }
}
```

[3 points] What will be the return value of fun(3,6)?

[4 points] Use big-O notation to express the running time of fun()? Explain your answer.

b) [3 points] What will be the output of the following code? Explain your answer.

```
enum direction {UP, DOWN, LEFT, RIGHT};

void foo(enum direction* a, enum direction b)
{
  enum direction* c = a;
  b = RIGHT;
  *c = LEFT;
  c = &b;
  *c = RIGHT;
}

int main(void) {
  enum direction d1 = UP;
  enum direction d2 = DOWN;
  printf("d1 = %d, d2 = %d\n", d1, d2);
  foo(&d1, d2);
  printf("d1 = %d, d2 = %d\n", d1, d2);
  return 0;
}
```

c) Consider the following function.

```
int foo(int n)
{
 if (n <= 0)
   return 0;
  return n+2 + foo(foo(n-2));
}
```

[2 points] What happens when foo is called with n = 2?

[2 points] What happens when foo is called with n = 3?

d) [3 points] Explain the functionality of the design pattern Singleton in C++? Give an example.

e) [3 points] Explain the difference between is-a and has-a relations in C++.

Problem 2 [20 points]
A Doubly Linked List is a linked list where each element has a pointer to the next element, as well as a pointer to the previous element.

```
struct DLL_node {
  int data;
  struct DLL_node* next;
  struct DLL_node* prev;
};
typedef struct DLL_node DLL_node_t;
```

a) [2 points] Write an algorithm that gets two doubly linked list nodes, and swap their values.

```
void swap(DLL_node_t* node1, DLL_node_t* node2) {

   }
```

b) [6 points] Implement in C the ***Insertion Sort*** algorithm on a Doubly Linked List.
The algorithm gets a pointer to the head of the list, and sorts the list.

```
void insertion_sort(DLL_node_t* head) {

}
```

c) [4 points] Consider the ***Quick Sort that choose the second element as a pivot***.
What is the running time of the algorithm when given as inputs a sorted array of n elements? Explain your answer.

d) [8 points] Write a function in C that solves the following problem.
Input: An array of ints A of length n with all values distinct such that for some (unknown) index K it holds that A[0…K] is sorted in an decreasing order, and A[K…n-1] is sorted in an increasing order.
Output: the index K.
The running time of the algorithm must be O(log(n)).
For example: A = [10, 8, 7, 5, 1, 3, 7, 9], the output should be 4 (i.e., A[4] = 1).

```
int findK(int* A, int n) {

}
```

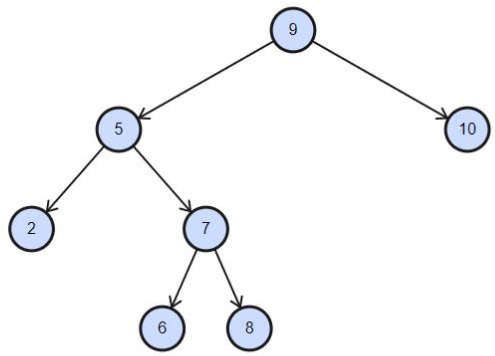
Problem 3 [20 points]

In this problem use the following struct for Binary Tree of ints.

```
struct BTnode {
  int value;
  struct BTnode* left;
  struct BTnode* right;
  struct BTnode* parent;
};
typedef struct BTnode BTnode_t;
```

a) [8 points] Write a function in C that gets a pointer to a node in a Binary Search Tree, and finds its successor. If the node is the maximal element in the tree, the function will return NULL.

In the example below the successor of 5 is 6, the successor of 8 is 9, and the successor of 9 is 10.

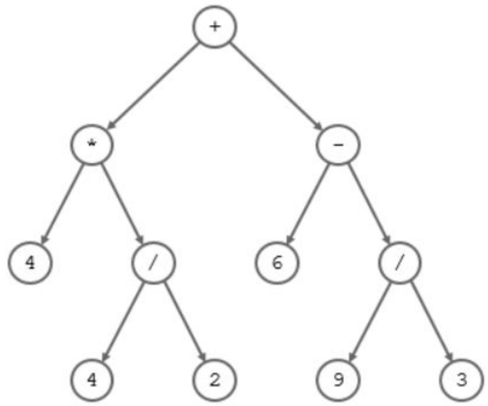



```
BTnode_t* get_successor(BTnode_t* node) {

}
```

b) [6 points] Write an algorithm that gets a Binary Tree representing an arithmetic expression, and prints the expression in Polish Notation. For example, for the tree below the function will print:

```
+ * 4 / 4 2 - 6 / 9 3
```



You may assume that the operations are implemented as

```
enum operators {PLUS='+', MINUS='-', MULTIPLY='*', DIVIDE='/'};

void print_prefix(BTnode_t* expression} {









}
```

c) [3 points] Convert the following expression from Polish Notation to the Infix Notation.

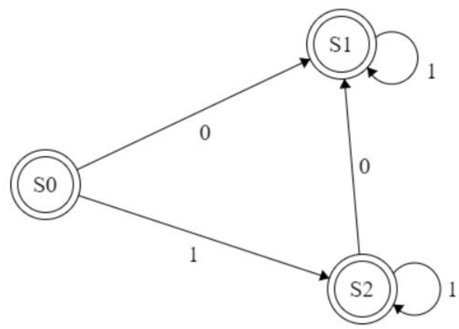
```
/ * - + 1 2 3 4 + 5 6
```

d) [3 points] Convert the following expression from Infix Notation to Reverse Polish Notation.

```
( ( ( 5 + 4 ) / ( 6 + 3 ) ) * ( 2 + 7 ) )
```

Problem 4 [20 points]

a) [5 points] Write a function in C that decides the language accepted by the following DFA. Explain your answer.




```c
int decide_lang(char* str) {





}
```

b) Consider the following regular expression: (a|b)* ((b*bb)|(a*a))

[4 points] Describe in words the language defined by the regular expression above.

[4 points] Draw a DFA that accepts the language defined by the regular expression.

c) Consider the following description of DFA:

$\sum = \{0,1\}$ $S = \{s_0, s_1, s_2\}$ $F = \{s_2\}$	$\delta(s_0,0) = s_0$ $\delta(s_0,1) = s_1$ $\delta(s_1,0) = s_2$ $\delta(s_1,1) = s_0$ $\delta(s_2,0) = s_2$ $\delta(s_2,1) = s_0$

[3 points] Draw the corresponding DFA.

[4 points] Describe the language accepted by the DFA.

Problem 5 [20 points - 4 points each item]
Implement the ADT *stack of ints*. The running time of each operation must be O(1).
In your code you may use the struct `node_t`.
If you want to use functions related to Linked List, you need to implement them.

```
struct node {
  int data;
  struct node* next;
};
typedef struct node node_t;
```

a)
```
typedef struct {



} stack_t;
```

b)
```
stack_t* stack_create() {



}
```

c)
```
void stack_push(stack_t* s, int item) {



}
```

d)
```
int stack_pop(stack_t* s) {



}
```

e)
```
int stack_is_empty(stack_t* s) {

}
```

Extra page