

CMPT125, Fall 2018

A sample of Final Exam

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 at 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.

```
void foo(unsigned int k, unsigned int n)
{
    for (int i = 0; i < n; i++)
        printf("%d ", k+i);
    printf("\n");

    if (k < n-1)
        foo(k+1, n);
}
```

[3 points] What will be printed when we invoke `foo(0, 3)`? Explain your answer.

[4 points] Use big-O notation to express the running time of `foo(0, n)` as a function of n ? Explain your answer.

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

```
enum direction {NORTH, SOUTH, EAST, WEST};

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

int main(void) {
    enum direction d1 = SOUTH;
    enum direction d2 = EAST;
    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 + foo(foo(n-1));
}
```

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

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

d) [2 points] What is a virtual method in C++? Give an example.

e) [4 points] Explain what is a class in C++ and what is an object in C++.

Problem 2 [20 points]

In this problem you need to use the following implementation of a Linked List on ints.

```
struct LLnode {
    int data;
    struct LLnode* next;
};
typedef struct LLnode LLnode_t;
```

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

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

}

b) [6 points] Implement in C the **Selection Sort** algorithm on a Linked List.

The algorithm gets a pointer to the head of the list, and sorts the list.

```
void selection_sort(LLnode_t* head) {
```

}

c) [4 points] Consider the **Insertion Sort** algorithm. How many swaps will it perform on the input $A = [5, 1, 8, 2, 6, 9]$? Explain your answer by writing intermediate steps of the algorithm if necessary.

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 $A[0 \dots K]$ is sorted in increasing order, and $A[K \dots n-1]$ is sorted in an increasing order, but $A[K] > A[K+1]$ and $A[0] > A[n]$. That is, $A[K+1, K+2, \dots, n-1, 0, 1, \dots, K]$ is a sorted array.

Output: the maximal element in the array.

The running time of the algorithm must be $O(\log(n))$.

For example: $A = [8, 10, 15, 1, 3, 5, 7]$, the output should be 15 (in this example $K = 2$).

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

```
}
```

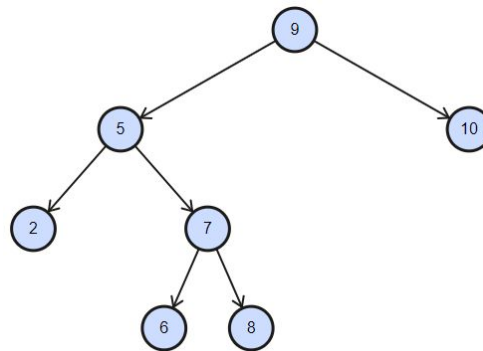
Problem 3 [20 points]

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

```
struct BTreeNode {  
    int value;  
    struct BTreeNode* left;  
    struct BTreeNode* right;  
    struct BTreeNode* parent;  
};  
typedef struct BTreeNode BTreeNode_t;
```

a) [8 points] Write a function in C that gets a pointer to the root of a Binary Search Tree, and a number, and returns the node containing this number.

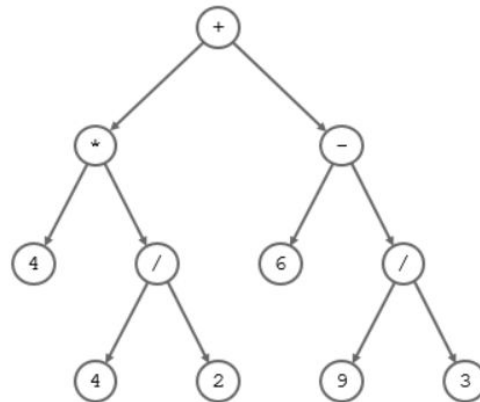
If the number is not in the tree, the function returns 0.



```
BTreeNode_t* find(BTreeNode_t* node, int num) {
```

```
}
```

b) [6 points] Write an algorithm that gets a Binary Tree representing an arithmetic expression, and prints the fully parenthesized expression in Infix 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='/'};
```

(Use %c to print an operator, and %d to print a number)

```
void print_infix(BTnode_t* expression) {
```

```
}
```

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

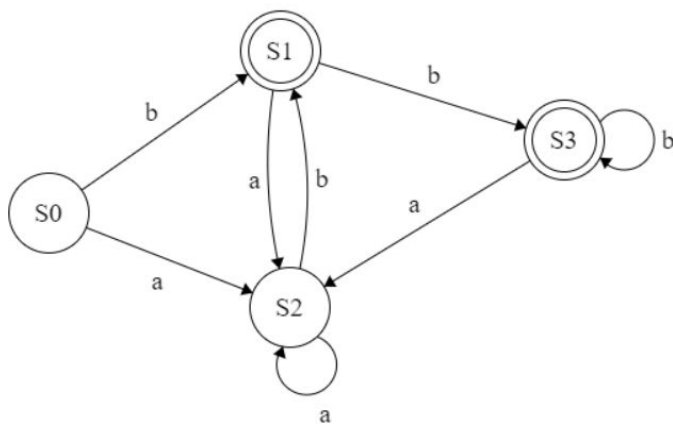
$(5 * ((3 - (2 / 1)) + 4))$

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

7 1 5 * - 6 3 / +

Problem 4 [20 points]

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



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


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

[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:

$\Sigma = \{a, b\}$	$\delta(s_0, a) = s_0$
$S = \{s_0, s_1, s_2\}$	$\delta(s_0, b) = s_1$
$F = \{s_2\}$	$\delta(s_1, a) = s_2$
	$\delta(s_1, b) = s_0$
	$\delta(s_2, a) = s_2$
	$\delta(s_2, b) = s_0$

[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 *queue 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 {`

`} queue_t;`

b) `queue_t* queue_create() {`

`}`

c) `void enqueue(queue_t* q, int item) {`

`}`

d) `int dequeue(queue_t* q) {`

`}`

e) `int queue_is_empty(queue_t* q) {`

`}`

Extra page

A large, empty rectangular box with a thin black border, occupying most of the page below the 'Extra page' header. It is intended for a drawing or additional text.