CMPT 125, Fall 2019

Final Exam
December 7, 2019

Name________________________

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

Problem 1	
Problem 2	
Problem 3	
Problem 4	
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 consists of four (4) problems. Each problem is worth 25 points.
4. Write your answers in the provided space.
5. There is an extra page at the end of the exam. You may use it if needed.
6. Explain all your answers.
7. Really, explain all your answers.

Good luck!

Problem 1 [20 points]

a) [4 points]Write the function str_reverse that gets a string and reverses it in place.

```
void str_reverse(char *str) {











}
```

b) [3 points] What will be the output of the following program?

```
#include <stdio.h>

enum colors {RED, GREEN, BLUE};

void foo(int* x, int *y, int z) {
    *x = z;
    x = y;
    *x = z;
    z = RED;
}

int main() {
    int a = RED, b = GREEN, c = BLUE;
    foo(&a, &b, c);
    printf("a = %d, b = %d, c = %d", a, b, c);
    return 0;
}
```

c) [3 points] Will the code below compile?
If yes, what will be the output? If no, explain why.

```
#include <stdio.h>

int main() {
  char str[10] ={‘a’,’b’,’c’,0,’1’,’2’,’3’,’0’,’\0’};
  char* ptr = str;
  printf("%s\n", ptr);
  return 0;
}
```

d) [4 points] What will be the output of the following code? Explain your answer.

```
int bar(int n) {
   if (n <= 1)
     return 1;

   int sum = 0;
   for (int i=1; i < n; i=i*2) {
     printf("%d ", i);
     sum += i;
   }
   printf("\n");

   return sum + bar(n/2);
}

int main(void) {
     printf("bar(8) = %d", bar(8));
     return 0;
}
```

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

e) [2 points] Explain what is void* in C.

Problem 2 [30 points - 3 points for each question]

Implement the ADT *Linked List of ints* so that the runtime of each operation is O(1).

a) Declare the type LLnode_t here:

b) Declare LL_t here:

c)
```
LL_t* create_LL() {








}
```

d)
```
void add_to_head(LL_t* list, int item) {








}
```

e)
```
int remove_from_head(LL_t* list) {








 }
```

Suppose you are given an ADT Doubly *Linked List of ints* `DLL_t` with the following operations.

```
    void add_to_head(DLL_t* list, int item);
    void add_to_tail(DLL_t* list, int item);
    int remove_from_head(DLL_t* list);
    int remove_from_tail(DLL_t* list);
    bool is_empty(DLL_t* list);
```

Suppose that the runtime of each operation is O(1).
Use DLL_t to implement stack of ints that also supports the stack_reverse operation.
The runtime of each operation must be O(1).

f)

```
typedef struct stack {



 } stack_t;
```

g)

```
void stack_push(stack_t* stack, int item) {



}
```

h)

```
int stack_pop(stack_t* stack) {



}
```

i)

```
bool is_empty(stack_t* stack) {


}
```

j)

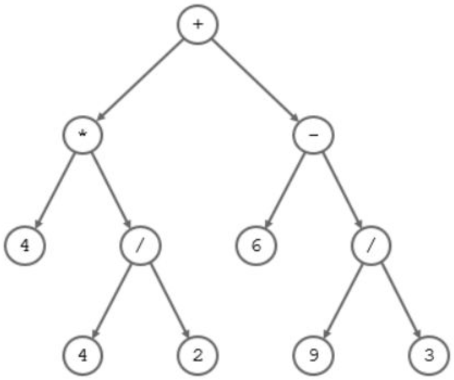
```
void stack_reverse(stack_t* stack) {



}
```

Problem 3 [25 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) [7 points] Write an algorithm that gets a Binary Tree representing an arithmetic expression, and returns the evaluation of the expression. For example, for the tree below the function will return 11. This is because (4 * (4/2)) + (6 - (9/3)) = 8 + 3 = 11.

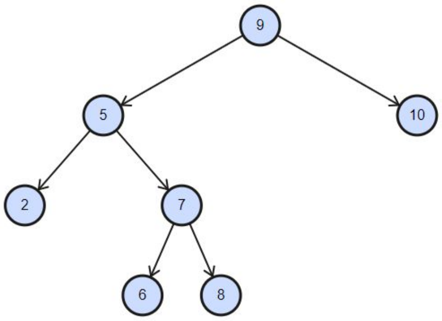


You may assume that the operations are implemented as

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

int evaluate(BTnode_t* expression} {









}
```

[4 points] Use the big-O notation to express the running time of evaluate. Explain your answer.

b) [10 points] Write a function that gets a pointer to the root of a Binary Search Tree, and returns the sum of the two smallest elements. For example, the smallest elements in the tree below are 2 and 5, and so the function should output 7. You may assume that the tree has at least 2 nodes.

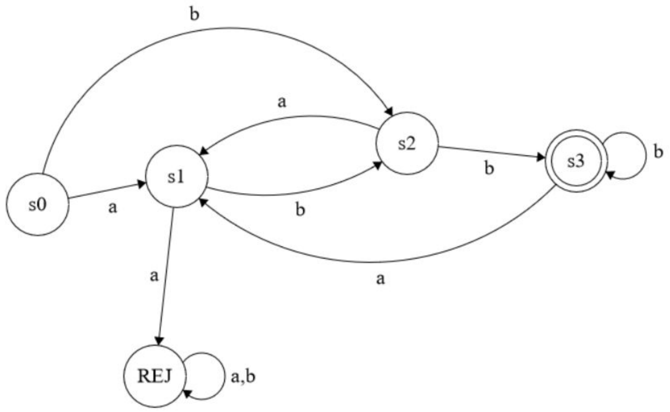


```
int* sum_of_min2(BTnode_t* root) {









}
```

[4 points] Use the big-O notation to express the running time of sum_of_min2. Explain your answer.

Problem 4 [25 points]

a) [4 points] Describe in words the language defined the following DFA. Explain your answer.



b) [5 points] Write a function in C that decides the language accepted by the DFA above.

```
bool decide_lang(char* str) {

}
```

b) Consider the following regular expression: a*(ba*ba*)*

[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:

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

[4 points] Draw the corresponding DFA.

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

Extra page