CMPT 125 D200, Spring 2022

Midterm Exam
February 28, 2022

Name_________________________

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

Problem 1	
Problem 2	
Problem 3	
Problem 4	
TOTAL	

Instructions:

1. Duration of the exam is 100 minutes.
2. Write your full name and SFU ID **clearly**.
3. This is a closed book exam, no calculators, cell phones, or any other material.
4. The exam consists of four (4) problems. Each problem is worth 25 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.
8. Really, explain all your answers.

Good luck!

Problem 1 [25 points]

a) [6 points] What will be the output of the following program? Explain your answer.

```
#include <stdio.h>
enum emph {BOLD, ITALIC, UNDERLINE};

int foo(int x, int *y) {
     int* z = &x;
     *y = *z;
     *z = -1;
     return x;
}
int main() {
     int a = BOLD, b = 1, c = 2;
     c = foo(a, &b);
     printf("a = %d, b = %d, c = %d", a, b, c);
     return 0;
}
```

b) [6 points] Will the code below compile?
If yes, what will be the output? If not, explain why.

```
#include <stdio.h>

int main() {
    char s1[20] = {'A','B',0,'C','D','E',0};
    char* str = s1;
    while (*str) {
      str = str+1;
    }
    printf("%s\n", str+1);
    return 0;

}
```

c) [7 points] Will the code below compile?
If yes, what will be the output? If not, explain why.

```c
#include <stdio.h>
#include <stdlib.h>

int* get_arr() {
  int* arr = malloc(3*sizeof(int));
  int* ret = arr;
  arr[0]=2;
  arr[1]=3;
  ret[2]=6;
  return ret;
}

int main() {
  int* a1 = get_arr();
  a1[0]=5;
  printf("a1 = [%d, %d, %d]\n", a1[0], a1[1], a1[2]);
  return 0;
}
```

d) [6 points] Let T(n) be the running time of foo(n). Use Big-O notation to express T(n).
Explain your solution.

```c
void foo(int n) {
  if (n>0) {
    int third = n/3; // if n is not divisible by 3, n/3 is rounded down
    foo(third);
    for(int i=third+1; i<=n ; i++)
      printf("i = %d ", i);
    printf("\n");
  }
}
```

Problem 2 [25 points]

a) [5 points] Consider the **Binary Search** algorithm. How many comparisons will it make on the input A = [1, 5, 7, 8, 20, 25, 30, 40, 60, 61, 62] when searching for 8? Explain your answer.

b) [8 points] Show an array with the values {1, 2, 3, 4, 5, 6, 7, 8} so that the **SelectionSort** makes swaps only in the last three iterations of the outer loop, and no other swaps.

c) [12 points] Implement the merge function that gets an array A of length n, and index mid, such that A[0,...mid-1] and A[mid…n-1] are sorted in the increasing order.
The function merges the two halves of A into a sorted array in time O(n).
* Note that some elements might be equal.
Remember to use malloc/free if you need to use a new array.

Explain your code if necessary.

```
void merge(int* A, int n, int mid) {

}
```

Problem 3 [25 points]

a) [8 points] Write a function that gets two strings and checks if str1 is the prefix of str2.

- is_prefix("abcd", "abcdef") returns true.
- is_prefix("a12b", "a12b") returns true.
- is_prefix("abcd", "ab") returns false.
- is_prefix("abcd", "KLM") returns false.

Explain your idea before writing code.

```
bool is_prefix(const char* str1, const char* str2) {















}
```

[4 points] What is the running time of your function? Use big-O notation to state your answer. Give the tightest possible answer.

b) [10 points] Implement the function str_find that gets two strings, *text* and *pattern*, and searches for the *pattern* as a substring of *text*. This function returns the index representing the beginning of the first appearance of the *pattern* in the *text*.
If *pattern* is not a substring of *text*, the function returns -1.
For example

- str_find("Hel**lo w**orld, Hello", "lo w") returns 3.
- str_find("aBaB**aBC**DaBC", "aBC") returns 4.
- str_find("Hello world", "wrd") returns -1.

You are not allowed to use any library functions to solve this, except for strlen().

```
int str_find(const char* text, const char* pattern) {

}
```

[3 points] What is the running time of your function in terms of the length of the strings in the worst case? Use big-O notation to state your answer. Give the tightest possible answer.

Problem 4 [25 points]

Consider the following function.

```
int what(unsigned int n) {
  if (n<=2)
    return n;
  int sum=0;
  for(int i=n/2; i<n; i++) // if n is odd, then n/2 is rounded down
    sum = sum+what(i);
  return sum;
}
```

a) [4 points] Compute what(4). Explain your answer.

b) [9 points] Rewrite the function what() with the same functionality so that on input n, it returns the answer in time O(n^2). Explain your answer.

c) [12 points] Write a function that gets an array of ints of length n, and returns an array of length n, such that output[i] is equal to the sum of the elements in the input subarray [0... i]. For example,

- input [1, 4, 3, 8, 2, 9]
- output [1, 5, 8, 16, 18, 27].

Make sure the returned array is allocated on the heap.

You may write helper functions if that makes the solution more readable.

- A correct answer with linear running time, will give you 15 points
- A correct answer with quadratic running time, will give you 10 points

```
int* sum_prefixes(const int* ar, int n) {

}
```

Extra page

Empty page