CpSc 2120 – Section 2, Fall 2021
Test 2 (March 29) CLOSED BOOK Total Points = 100

Question 1: (15 points)

(A) Given the declaration “int x[10];”, which of the following statements will correctly read an integer value into x[2]?
(a) fscanf(stdin, “%d”, *x+2);  (b) scanf(“%d”, *x[2]);  (c) scanf(“%d”, x+2);
(d) fscanf(stdin, “%d”,x[2]);  (e) none of them.

(B) Consider the statement: "My salary was increased by 15%!" Circle the one that will EXACTLY reproduce the statement.
(a) printf("My salary was increased by 15\%
\n");
(b) printf("My salary was increased by 15%!\n");
(c) printf("My salary was increased by 15'\%
\n");
(d) printf("\My salary was increased by 15%\%!\n");
(e) printf("\My salary was increased by 15%\n");

(C) Consider the code:

```c
double x = -3.5, y = 3.5;
printf( "%.0f : %.0f
\n", ceil( x ), ceil( y ) );
printf( "%.0f : %.0f
\n", floor( x ), floor( y ) );
```

What will the code above print when executed? Circle the correct one;
Note: ceil => rounds up 3.2 to 4 floor => rounds down 3.2 to 3

<table>
<thead>
<tr>
<th></th>
<th>(a) -3 : 4</th>
<th>(b) -4 : 4</th>
<th>(c) -4 : 3</th>
<th>(d) -4 : 3</th>
<th>(e) -3 : 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-4 : 3</td>
<td>-3 : 3</td>
<td>-4 : 3</td>
<td>-3 : 4</td>
<td>-4 : 4</td>
</tr>
</tbody>
</table>

(D) Consider the declaration: char txt[20] = "Hello world!0";
How many bytes are allocated by the declaration?
(a) 11  (b) 12  (c) 13  (d) 20  (e) 21

(E) Consider the declaration
typedef struct{int x[3], float y, int z[5];}mystery; mystery toy;
How many bytes are allocated to store the variable toy?
(a) 30  (b) 35  (c) 36  (d) 40  (e) 9  (f) none
Question 2 (12 points): An arithmetic expression is called fully parenthesized if and only if each operator (+, −, ×, ÷) is parenthesized along with its two operands. For example, the expression \(a+b/c\) is fully parenthesized when written as \((a + (b/c))\) or the expression \(a+b+c – d \times e\) is written as \(((a+b) + c) – (d \times e)\) and so on. Write a complete program that will accept a fully parenthesized arithmetic expression (consisting of the 4 operators and single digit integers), evaluate the expression and printout the value on the terminal. Example: `a.out ((5+6) \times 8)` will print 88. Assume the input expression does not have any blanks or any other symbol. Note: You can store the input string using any data structure you choose, but you cannot scan the string more than once in your program.
**Question 3 (2+2+2+2+2 = 10 points):** Circle one answer:

(a) Suppose a BST is traversed in the following order recursively: Right, root, left. The output sequence will be

   (A) Ascending Order   (B) Bitonic Sequence   (C) No specific order   (D) Descending order

(b) The pre-order and post order traversal of a Binary Tree generates the same output. The tree can have maximum

   (A) Three nodes   (B) Two Nodes   (C) one node   (D) Any number of nodes

(c) The ideal goal of hashing is to produce a search that takes

   (A) (1) time   (B) O(n) time   (C) O(n log₂n) time   (D) O(log₂n) time

(d) A characteristic of the data that binary search uses but the linear search ignores is the_________.

   (A) Order of the elements of the list.   (B) Length of the list.   (C) Maximum value in list.   (D) Type of elements of the list.

(e) An ADT is defined to be a mathematical model of a user-defined type along with the collection of all __________ operations on that model

   (A) Cardinality   (B) Assignment   (C) Primitive   (D) Structured

**Question 4 (7 points)**

Write a complete C or C++ program that will accept a user provided string of characters (numerals and/or alphabets, no special characters) – an example execution will look like a.out ac4dfg7. Your program will analyze the string and will generate an appropriate message on the terminal if the input string is a palindrome or not [a string is a palindrome if it reads the same forward and backward] – a.out ac4dfg7 will generate ac4dfg7 is not a palindrome, while a.out 3d3d3 will generate 4z3z4 is a palindrome.
Question 5 (6+2+2+2+2 = 14 points):
   a) Consider the following arbitrary arrival of integers: {30, 40, 50, 70, 72, 60, 35, 45} in that order. Starting from an empty binary search tree, insert the integers one at a time. Show the resulting binary search tree the usual way.

   b) Then, delete the node 35; show the resulting BST.

   c) Then delete the node 50; show the resulting BST.

   d) Then insert 20, 25 and 29 in that order one at a time; show the BST.
e) Then delete node 30; show the BST.

Question 6 (6 points):

a) What is the output of the following program? Circle one.
   int a = 35, b=50; int *p1=&a, *p2 =&b;
   p1=p2; printf("%d %d\n", *p1, *p2); printf("%d %d\n", a, b);
   (1) 35 50 (2) 50 50 (3) 35 50 (4) 50 50

b) Consider the following program segment; what is the output? Circle one.
   void fun(int *a, int b){int c; a=a+100; c=b; b=999;}
   int d=10, e =20; fun(&d, e); printf ("%d %d\n", d, e);
   (1) 10 20 (2) 110 20 (3) 110 999 (4) none of those.

c) What values will be printed by the following program segment?
   int proc (int x){x+= 2; return x};
   int x= 8; int y = proc(x); printf("%d %d\n", x, y);
Question 7 (5+5+10= 20 points):

Consider the following recursive function, assuming $0 \leq m \leq n$ and $n \geq 1$.

```c
int fun (int m, int n){
    if (((n == 1) || (m == 0) || (m == n))
        return (1);
    else return (fun (n - 1, m) + fun (n - 1, m - 1)); }
```

(a) What are fun(4,2)? fun(5,3)? fun(6,4)? fun(8,3)? fun(9,2)?

(b) What does this function do given any $m$ and $n$ within the constraints? Compute the function for some smaller values of $m$ and $n$; try to generalize; observe that the recursion ends in finite time; observe the similarity with how we wrote the recursive function for Fibonacci numbers in class; and then give a precise one sentence description of the purpose of the function.

(c) Then, redesign the function in an iterative way removing the recursion.
**Question 8 (10 points):** Consider the following arbitrary array of integers: \{50, 4, -1, 2, 8, 7, 12, 10, -5, -6, 9, 16, 3, -3, 49, -9\}. We want to sort the array in ascending order by using Mergesort (we have done this in class). Show the array in successive stages (after each time “merging” is executed, as shown in the class example) of your algorithm.

<table>
<thead>
<tr>
<th>initial</th>
<th>50</th>
<th>4</th>
<th>-1</th>
<th>2</th>
<th>8</th>
<th>7</th>
<th>12</th>
<th>10</th>
<th>-5</th>
<th>-6</th>
<th>9</th>
<th>16</th>
<th>3</th>
<th>-3</th>
<th>49</th>
<th>-9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Question 9 (2+2+2 = 6 points):** Define precisely what a min-heap is. (a) Consider the array A[9] = \{5, 6, 7, 12, 14, 16, 17, 20, 3\}. Does this array represent a min-heap? Explain properly with a “mental picture” of the array as a binary tree; (b) is any valid heap always a binary search tree? If so, explain why so and if not, why not? Explain precisely.
Bonus Question (4+4+4 = 12 points):

(a) In how many ways can 6 people sit around a round table if 3 of the people – Jenny, Tom, and Melissa – all want to sit together? Now, Tom and Melissa want to sit together while 3 of those still sit together. What is the number under this additional restriction? Justify your answer [No credit without explanation; exhaustive enumeration does not count]

(b) There are 2 Senators from each of the 6 states – SC, NC, LA, TN, NY, CO. We wish to make a 3-Senator committee in which no two members are from the same state. In how many ways can the committee be formed such that no 2 Senators are from the same state? Justify your answer [No credit without explanation; exhaustive enumeration does not count.]

(c) The L-shape pictured is formed from three squares, each 1 cm on a side. Five of these L-shapes are placed next to each other to form a figure. What is the least possible perimeter of the figure they form, in cm?