IT3312E Data structures and Algorithms Winter 2019 Final Exam – 90 minutes	1

Name:_____**Student ID:**_____

Question 1 [10 points]:

1. Which of the following is/are an example(s) for a postfix expression?
   a) a*b(c+d) b) abc*+de c) +ab d) a+b-c
2. Give one advantage of a Linked List, as compared to an array

Question 2 [20 points]:

a. Represent the tree in Figure 1. by using Left-most child, right-sibling representation

b. List the nodes on the tree when we traverse the tree in post-order

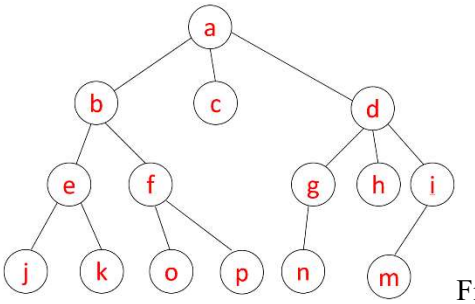


Figure 1.

Question 3 [20 points]: Using the function BUILD-MAX-HEAP in heap sort to create a max-heap for the array:
14, 11, 13, 12, 26, 19, 20, 24, 18, 27

Question 4 [30 points]:

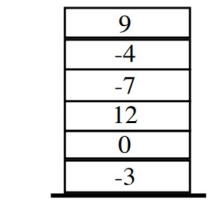
a. Binary search [10 points]: Given an array ***A*** consisting of 17 integer numbers that are sorted in ascending order:
16, 19, 25, 30, 40, 45, 78, 90, 110, 140, 170, 200, 205, 210, 290, 300, 305
We want to determine whether the ***target*** = 211 is appeared in the array ***A*** by using the binary search function `int binarySearch(int low, int high, int A[], int target).`
Therefore, the first call is `binarySearch(0, 16, A, 211);`
How many times this function are called in total, and list all the remaining calls.

b. Binary search tree [10 points]: Draw the binary search tree that was built by inserting the following data values in the order given: 42, 17, 89, 53, 72, 91, 3, 88

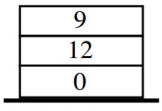
c. [5 points] List the data values that would be examined in a search of the binary search tree for the value 53.

d. [5 points] Given the binary search tree obtained in the question b) above, carry out the following operations in sequence: add 5, delete 53, delete 89.

Question 5 [20 points]: Complete the definition of the function **`void removeNegatives(Stack numStack)`**, which removes all of the negative numbers from a stack of numbers. The remaining numbers should be left in the same relative order.
Note you may only use the standard Stack methods (push, pop, empty, top) to manipulate the Stack.



`numStack` before calling `RemoveNegatives(numStack)`



`numStack` after calling `RemoveNegatives(numStack)`

Question 1.

```
typedef struct listNode * listPointer;
typedef struct listNode {
   int data;
   listPointer next;
};
```

Complete the following function **listPointer concatenate(listPointer list1, listPointer list2)** to concatenate two circularly singly linked list into one circularly singly linked list.
The function produces a new list which contains list1 followed by list2. It returns the pointer which points to the new list. The arguments list1 and list2 point to the tails of the lists.

Question 2. There is a list of students consisting of their information: name, year of birth (integer number), and grades for two courses IT3312E and IT3315E (2 integer numbers).

- **a.** Declare the *doubly linked list* used to represent this list with the pointer **head** pointed to the first element and pointer **tail** pointed to the last element.
- **b.** Using the above implementation and the **tail** pointer, complete the method **`boolean Contain(int c)`** returns true if there is a student in the list having age equals to ***c***; false otherwise.

Question 3. Given the following program:

Program 1

```
1. #include <stdio.h>
2. int f(int a[], int n){
3.    if(n <= 0) return 0;
4.    return a[n-1] + f(a,n-1);
5. }
6. int main(){
7.    int a[7] = {9,6,7,2,8,3,1};
8.    printf("f = %d\n",f(a,7));
9. }
```

The function **`f`** (lines 2—5) with arguments: array **`a`** having **`n`** elements (elements are indexed from 0 to **`n-1`**).

a) What does the function **f** do? Given the time complexity as a function of **n**.
b) Explain and show the results displayed on the screen when running the Program 1 above.

Question 4. Given a parenthesized expression, test whether the expression is properly parenthesized
Examples: {a*[b+c]-([c*e]+(d-e))}+f*[c-g] is proper

[a+{b*[c/(d-e)]}+(d/e) is not proper

Using stack, fill in the Table 1 all steps you need to do to check whether the following expression is proper or not: {a*{((b+c)-e-([c*e]+(d-e)))+f*[c-g]}

Step	Element of expression	Operation performed	Stack (bottom to top)
1	{	Push { to stack	{
2	a	Ignore	{
3	*	Ignore	{

Table 1. Stack application: Parentheses Matching

Question 5.

A storage table is allocated *m* slots with indexes 0, 1, …, *m*-1 to store key-value pairs. Initially, the table is empty. Draw the table state (each position only needs to write the key if in use and leave it blank if not) when inserting keys 22, 1, 13, 11, 24, 33, 18, 42, 31 respectively into the table with the $m = 11$.

a) The hash table applies the open addressing method with the hash function $\boldsymbol{h(k,i) = (k \bmod m + i) \bmod m}$

b) The hash table applies the open addressing method with the double hash functions: $\boldsymbol{h(k,i) = (h_1(k) + ih_2(k)) \bmod m}$

- $h1(k) = k \bmod m$
- $h2(k) = 1 + (k \bmod (m - 1))$