UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

*Academic Year **2020** – 2nd Year Examination – Semester 4*

IT4105 – Programming II

Part 1 - Multiple Choice Question Paper

(ONE HOUR)

Important Instructions:

- The duration of the paper is **1 (one) hour**.
- The medium of instruction and questions is English.
- The paper has **25** questions in **11** pages.
- All questions are of the MCQ (Multiple Choice Questions) type.
- All questions should be answered.
- Each question will have 5 (five) choices with **one or more** correct answers.
- All questions carry equal marks.
- There will be a penalty for ***incorrect responses*** to discourage guessing.
- The mark given for a question will vary from 0 (*All the incorrect choices are marked & no correct choices are marked*) to +1 (*All the correct choices are marked & no incorrect choices are marked*).
- Answers should be marked on the special answer sheet provided.
- Note that questions appear on both sides of the paper. If a page is not printed, please inform the supervisor immediately.
- Mark the correct choices on the question paper first and then transfer them to the given answer sheet which machine will be marked. **Please completely read and follow the instructions given on the other side of the answer sheet before you shade your correct choices.**
- Calculators are **not** allowed.
- All Rights Reserved.

1) There exists an array implementation of a circular queue called “queue”. The length of the queue is 8 elements and elements are filled up to queue [6] with the remaining locations left unfilled. Where does the enqueue method place three new entries in the queue? Assume that the staring index of the array is 0.

(a) queue [7], queue [8], queue [9]
(b) queue [7], queue [8], queue [0]
(c) queue [7], queue [1], queue [2]
(d) queue [7], queue [0], queue [1]
(e) none of above.

2) Consider the following pseudocode segment.

```
int coeff(n,k)
    if (k= 0 or k = n)
        return 1;
    else
        return (coeff(n-1,k-1) + coeff(n-1,k));
```

What is the return value of coeff(5,2) for the above pseudocode segment?

(a) 5	(b) 2	(c) 15
(d) 10	(e) 20	

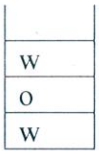
3) Assume there exists a HashMap data structure which is filled with n elements. What are the complexities of searching an element from the hash table in the average-case if hash collisions exist in the above-mentioned hash table?

(a) O(n)
(b) $O(\log_n)$
(c) O(1)
(d) $O(n\log_n)$
(e) $O(n^2)$

4) If a hash table does not carry any hash collisions, what is the worst-case complexity of deleting an element from the hash table?

(a)O(n)	(b)O (1)	(c)$O(n^2)$
(d)$O(n \log_n)$	(e)$O(\log_n)$	

5) You are given a stack data structure where the current content of the stack is as follows:

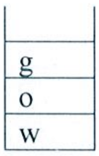
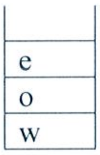
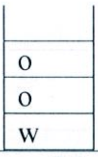
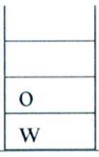


Consider the following pseudocode segment,

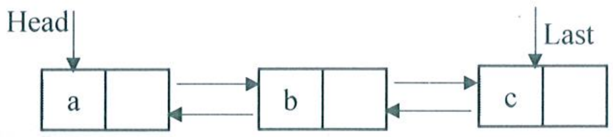
```
function stack_computation(String input)
{
        While(character stream is not empty)
        {
                c = read a character from input
                if ( c == "a" || c == "e"|| c == "i"|| c ==
"o"|| c == "u")
                {
                        pop a character
                }
                else
                {
                        push c on the stack
                }
        }

}
```

What will be the final content of the stack for the function `stack_computation ("ago")`

(a) 	(b) 
(c) 	(d) 
(e) None of the above	

6) Suppose you are given a doubly linked list in which the Head and Last pointers indicate the first and the last elements of the list. Next and Prev pointers are used to traverse the linked list with forward and backward pointers, respectively.



What are the steps involved in adding a new_node as the head of the doubly linked list?

```
(a)
Node new_node = new_node(data)
new_node.next = Head
new_node.prev = null
Head.prev = new_node
Head = new_node
```

```
(b)
Node new_node =
new_node(data)
new_node = Head.next
new_node.prev = null
Head.prev = new_node
Head = new_node
```

```
(c)
Node new_node = new_node(data)
new_node.next = Head.next
new_node.prev = null
Head = new_node.prev
Head = new_node
```

```
(d)
Node new_node =
new_node(data)
new_node.next =
Head.prev
new_node.prev = null
Head.prev = new_node
Head = new_node.next
```

```
(e)
Node new_node = new_node(data)
new_node.next = Head.next
new_node.next = null
Head.next = new_node
Head = new_node
```

7) Consider the following pseudocode,

```
void func(int n)
{
    Stack Smap;  //stack is initialized to empty
    while (n > 0)
    {
        if(!n%2)
        {
            push(n);
        }
        n++;
    }

    while (!isEmpty(Smap))
      print("%d ", pop());
}
```

What is the functionality of the above pseudocode function?

a) Prints the value of log n.
b) Prints the even values of n
c) Prints the odd values of n.
d) Prints even values of n in reverse order.
e) Prints odd values of n in reverse order.

8) You are given a priority queue data structure. What is the time complexity of searching the highest priority element from the queue?

(a) O(n)	(b) O (lg n)	(c) O (1)
(d) O(n^2)	(e) None of above	

9) What statement/s is/are true regarding adjacency list/ matrix representations of a graph G (V, E), where V and E are considered to be vertices and edges respectively.

a) Look up of a node in adjacency matrix representation requires you to traverse the entire row and column of the matrix where node resides.
b) Size of the adjacency matrix is |V|
c) When visiting edges in adjacency list that starts with node v, traversing entire linked list of node V for linked lists is a must.
d) Complexity of visiting an edge in the average case is Θ (size_of_linked_list(V)) for adjacency lists, while an adjacency matrix provide faster lookup where complexity for lookup is Θ(1)
e) Visiting all elements of adjacency matrix shows complexity of Θ (|V|).

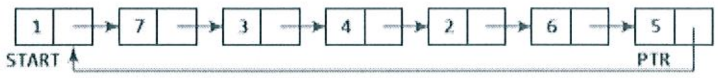
10) Consider the statements below. Note that V and E are considered to be vertices and edges, respectively.

(i) Both (Depth First Search (DFS) and Breadth First Search (BFS) can be used to find it the undirected graph is connected.
(ii) Complexity of finding that an undirected graph is connected, will be O(V + E).
(iii) Breadth First Search (BFS) can be used to find connected components of a graph.
(iv) Complexity of finding connected components will be O(V + E).

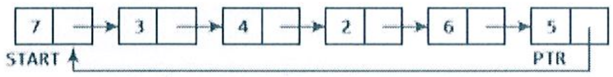
Which of the following are correct related the graphs?

(a) Only (i) is correct	(b)Only (iii) is correct	(c)Only (i) & (ii) correct
(d)All are Correct	(e)(i), (ii),(iii) and (iv) are correct	

11) Consider the following circular singly linked list with the given pointers.



If one deletes the first node from the above list, the resultant list with pointers is given below.



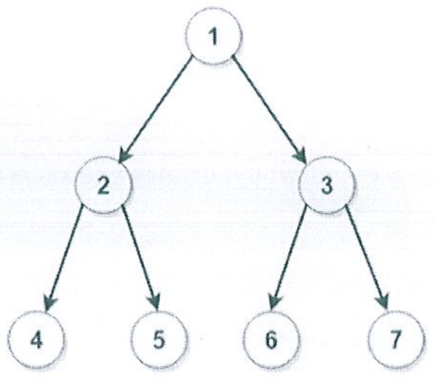
What are the correct pseudo commands used to perform the above deletion?

(a)
`KILL START     // REMOVE START`
`PTR.NEXT=START.NEXT`
`PTR=START`
	(b)
	`START.NEXT=PTR`
	`PTR.NEXT=START`
	`KILL START     // REMOVE START`

(c)
`PTR.NEXT=STRAT`
`START=START.NEXT`
`KILL START     // REMOVE START`
	(d)
	`PTR.NEXT=START.NEXT`
	`KILL START     // REMOVE START START=PTR.NEXT`

(e)
`START.NEXT=PTR.NEXT`
`KILL START     // REMOVE START`
`START=START.NEXT`	

12) Consider the following binary tree.



Which of the following statement(s) is/are valid in connection with the above tree?

(a) It is a complete binary tree (b) It satisfies the binary search tree properties (c) In-order traversal gives the increasing order of node values. (d) It's height is 2 (e) It has 3 terminal nodes.

13) A programmer wants to create two binary trees inserting the following nine (09) nodes in the given order based on the following conditions:

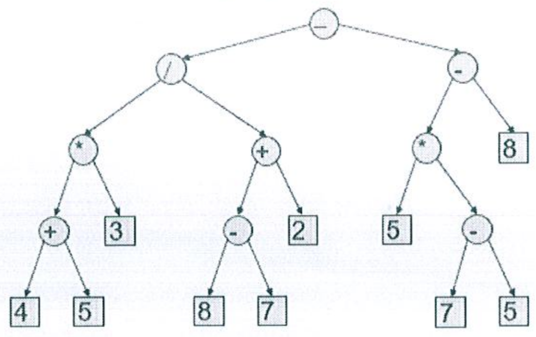
(i) To indicate the minimum depth of node G
(ii) To indicate the maximum depth of node G

What is the (i) minimum depth of the Node G and (ii) maximum depth of node G in the above-created trees respectively?

Nine (09) nodes are: A, B, C, D, E, F, G, H, I

(a) (i) 3 and (ii) 4	(b) (i) 2 and (ii) 6	(c) (i) 3 and (ii) 5
(d) (i) 3 and (ii) 6	(e) 2 and (ii) 7	

14) Consider the following expression tree.



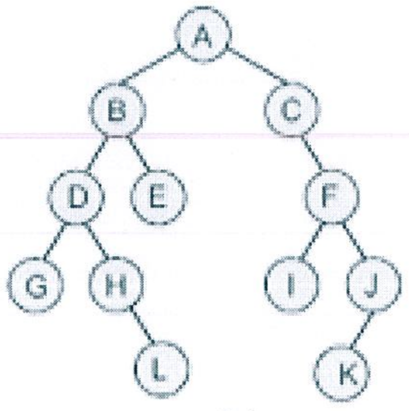
Which order of traversal can be used to obtain the postfix expression from the above tree?

(a) In-order Traversal
(b) Depth first Traversal.
(c) Breadth First Traversal.
(d) Post-order Traversal
(e) Pre-order Traversal

15) Which of the following is/are false about a binary search tree?

(a) The left child is always lesser than its parent
(b) The right child is always lesser than its parent
(c) The left and right sub-trees should also be binary search trees.
(d) Post-order sequence gives increasing order of elements
(e) One of the major applications in Binary Search Trees is dictionary implementation

16) Consider the following tree.



Which of the following shows the Breadth-First Traversal (BFT) node order of the above tree?

(a) A, B, C, D, E, F, G, H, I, J, K, L
(b)A, B, C, D, E, F, G, H, I, J, L, K
(c) L, K, G, H, I, J, D, E, F, B, C, A
(d) A, B, D, E, G, H, L, C, F, I, J, K
(e) None of the above

17) Consider the following pseudocode algorithm segment.

```
fmdMin(t)
 if (t = = null)
        return null;
 else if (t->left= = null)
            return t;
 return findMin(t->left);
 }
```

Which of the following tree(s) is/are suitable to find the minimum value from the above pseudocode algorithm segment?

(a) Pointer based implementation of binary trees
(b) Pointer based implementation of binary search trees
(c) Pointer based implementation of AVL trees
(d) General trees
(e) A complete binary tree

18) For a given undirected graph G having v vertices and e edges which are connected and has no cycles, which of the following statement(s) is/are true?

(a) v> e	(b) v=e-1	(c) (i) v=e
(d) e=v+1	(e) v=e+1	

19) Suppose you have a directed graph representing all the functioning bus routes among the cities. What algorithm might be used to find the best sequence of routes from one city to another?

(a) Depth first search algorithm
(b) Topological sort algorithm
(c) Breadth first search.
(d) Level order traversal algorithm.
(e) A shortest-path algorithm

20) Which of the following is/are (an) advantage(s) of an adjacency list representation over an adjacency matrix representation of a graph?

(a) Deleting a vertex in adjacency list representation is easier than in adjacency matrix representation.
(b) For sparse graphs, memory can be saved in adjacency matrix representation than the adjacency list representation.
(c) pointer-based implementation of adjacency list is easier than the adjacency matrix representation.
(d) (a), (b), and (c) are correct
(e) (a), (b) and (c) are wrong

21) What is the time complexity of the following code?

```
     public int fun(int n) {
   int sum = 0;
   for (int i = 0; i < n; i++)
       for (int j = i+1; j < n; j++)
           for (int k = 1; k < n; k = k*2)
                   if (a[i] + a[j] >= a[k])
                       sum++;
     }
```

(a) 11n+ 5 lg(n) + 100	(b) O(3 n^2)	(c) O(n lg n^2)
(d) O(n^2 lg n)	(e) O(n^2)	

22) Asymptotic notation is used to describe the time complexity of algorithms and this notation requires the calculation of the number of operations as a function of input size. Given below are the number of operations performed by different functions when the input size is "n". Which of the following function(s) can be represented using O(n^3)?

(a) 11n+ 5 lg(n) + 100	(b) 50 n^2	(c) 20000 n^3
(d) n	(e) All of (a), (b),(c) and (d)	

23) Two students were asked to develop an algorithm to solve a given problem. The algorithm developed by the first student performs **"n lg(n)"** operations and the second student's algorithm performs **"n^2"** operations, when input size is **"n"**. If the input size is 1 million (10^6), approximately how much faster is the solution given by the first algorithm compared to the second one?

(a) 20 times
(b) 1000 times
(c) 50000 times
(d) 1000000 times
(e) 20000 times

24) You are given the following array. How many comparisons are required to search it using the binary search algorithm when the target value is equal to 35?

[8, 15, 25, 37, 35, 41, 61, 88, 92]

Note: Set top=arraysize – 1; set bottom = 0; and when calculating the middle, set middle to the ceiling of (top+bottom)/2.

(a) 4	(b) 1	(c) 2
(d) 3	(e) 5	

25) The following array needs to be sorted in ascending order. If Heap Sort is used to sort the array, what statement(s) is/are correct?

[31, 18, 89, 46, 53, 25, 76]

(a) Max-heap is created and 89 is swapped with 46.
(b) Max-heap is created and 53 is swapped with 25
(c) Max-heap is created and 53 is swapped with 18.
(d) Max-heap is created and 46 is swapped with 25.
(e) Max-heap is created and 89 is swapped with 76.

****