


Faculty of Engineering

SCIT

CSE

III SEM. B.Tech

ODD Semester End Term Examination 2020-21

CS 2103, Data Structures and Algorithms

Time: 01 Hour 15 mins.**MAX.MARKS: 20***Instructions to Candidates*

- Answer all five questions in your own words.
- Missing data, if any, may be assumed suitably.

Q. No.		Marks	CO Mapping
1.	<p>Complete the code segments below so that the values stored in a given single dimensional array of integers of size 'n' gets swapped with its index position. For example: if the value at arr[i]=x then contents at arr[x] shall become i i.e., arr[x]=i.</p> <pre> # include <stdio.h> void swap(int arr[], int n) { // _____ Fill code Segment 1.1 } void printArray(int arr[], int n) { // _____ Fill code segment 1.2 } int main() { // _____ Fill code segment 1.3 printArray(arr, n); swap(arr, n); printf("Modified array is \n"); printArray(arr, n); return 0; } </pre>	3	CO1

2.	<p>Complete the code Segments 2.1 and 2.2 given below to encode a string stored using singly linked list in the following fashion:</p> <p>The input string is a sequence of character and number (single digit). If the first element of linked list is 'b' and the next element is '7' then '7' shall be replaced with 'i', which is an alphabet which comes at seventh position after 'b'. Similarly, if the element of the Linked List is 'z' and the next element is 3 then 3 shall be replaced with 'c'.</p> <p>Write only required code segment mentioning proper segment number.</p> <p>Example: Input: b7m4e5 start->b->7->m->4->e->5->z->3->NULL Output: bimqej start->b->i->m->q->e->j->z->c->NULL</p> <p><u>Code Segment:-</u></p> <pre>#include <stdio.h> #include <stdlib.h> #include <ctype.h> struct node { char data; struct node *next; }; struct node * createLL (struct node * start) { // _____ <u>code segment 2.1</u> } void print(struct node *start) { for(; start!=NULL ; start=start->next) printf(" %c", start->data); } struct node* convert (struct node *Start) { // _____ <u>code segment 2.2</u> } void main() { struct node *Start1=NULL, *Start2=NULL; Start1=createLL(Start1); print(Start1); Start2=convert(Start1); print(Start2); }</pre>	4	CO1
----	---	---	-----

3.	<p>For servicing purpose, a garage has only one gate for entry and exit and can accommodate 'M' cars at a particular time. Each of the car is identified by its car-id (positive integer) and each car has its own servicing time (Time to Service that Car). When a car enters the garage, the cars needs to be rearranged as per the service time and the car with the minimum service time shall always be close to the gate and once serviced shall be allowed to exit.</p> <p>Keeping in view the above scenario, Complete the code segment 3.1, 3.2 and 3.3 using singly linked list (Write code segment only)</p> <pre> #include <stdio.h> #include <stdlib.h> #include <conio.h> struct node { int car_id, sr_time; struct node*next; }; struct node *head=NULL; ... enter_garage(.....) // code segment 3.1 { } ... exit_garage() // code segment 3.2 { } void main() { int choice, car_id, time; do { printf("1.Car Entry in Garage for Service\n2. Car Exit from Garage –Service Completed\n 3. Exit"); scanf("%d", &choice); switch(choice) { case 1: printf("enter car_id and service time") scanf("%d%d", &car_id, &time); enter_garage(car_id,time); break; case 2: x=exit_garage(); printf("The removed car id is %d ", x); break; case 3: exit(0); } } while(choice>=1 && choice<=3); getch(); } </pre>	4	CO1, CO3 and CO4
----	---	---	---------------------------

4.	<p>Write a non-recursive function to count number of leaf nodes in a binary search tree. The function prototype for counting the number of leaf nodes in a binary search tree is as below:</p> <pre>int countLeaf(NODE * root);</pre> <p>Assume that there exists a built-in data structure STACK, push and pop operations are already defined. The prototype of push and pop functions are as below:</p> <pre>void push(STACK *s, NODE *p); NODE * pop(STACK *s);</pre>	3	CO2 &CO3
5.	<p>For the graph given below:</p> <pre> graph TD A --- 3 B A --- 5 C B --- 2 C B --- 5 D B --- 6 E C --- 7 F D --- 4 E D --- 2 G E --- 6 F E --- 7 G </pre> <p>(i) Give the breadth first traversal of the graph with starting graph node as A. Show traversal steps.</p> <p>(ii) Give the depth first traversal of the graph with starting graph node as A. Show traversal steps.</p> <p>(iii) Construct the minimum spanning tree using Kruskal Algorithm and also compute the total weight of the spanning tree. Show the spanning tree construction steps.</p> <p>(iv) Construct the minimum spanning tree using Prims Algorithm and also compute the total weight of the spanning tree. Show the spanning tree construction steps</p>	6	CO2