



# MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

## III SEMESTER B. TECH (COMPUTER SCIENCE & ENGINEERING)

MID SEMESTER EXAMINATION, OCTOBER 2020

SUBJECT: DATA STRUCTURES & APPLICATIONS (CSE 2152)

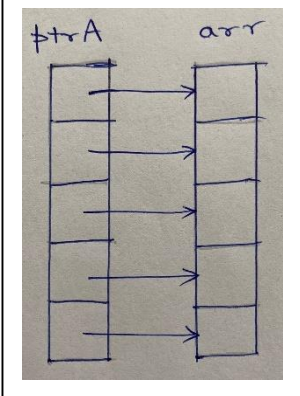
REVISED CREDIT SYSTEM

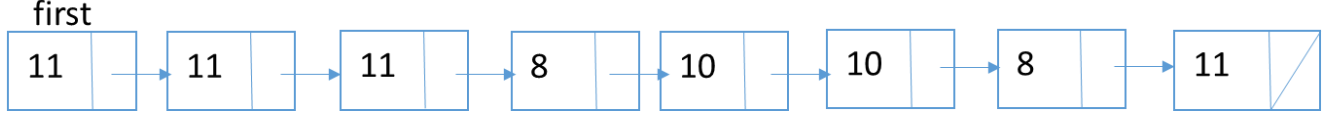
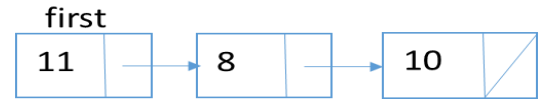
Date of Exam: 22/10/2020

Time: 90 Minutes

MAX. MARKS: 20

**Note: Answer ALL the questions.**

1	<p>Write a complete C program to do the following. Read an integer 'n' from the keyboard. Read 'n' integers &amp; store in an array 'arr'. Create an array of pointers to integers, 'ptrA'. Each pointer in 'ptrA' should point to the corresponding element in 'arr' as shown in figure. Write function <b>display()</b> to display the elements of 'arr'. Also write a function <b>rotate()</b>, which rotates the elements in 'arr' one position to the right without changing the position of the elements in the array, that is, only the pointers have to be rearranged. For both the functions, the only parameters that may be passed are 'ptrA' and 'n'.</p> <p><u>Example:-</u> Suppose array 'arr' contains the elements 1, 2, 3, 4, and 5. First time when display() is called the output should be "1 2 3 4 5". When display() is called, after rotate(), the output should be "2 3 4 5 1".</p>		3								
2	<p>Consider the following program. How many times will the word "fibonaci" be printed? What will be printed in the last line of the output?</p> <pre>int fix(int n){     if(n==0) return 0;     if(n==1) return 1;     else{         printf("fibonaci\n");         return (fix(n-1) + fix(n-1));     } }</pre> <pre>void main(){     int num, fnum;     num=5;     fnum=fix(num);     printf("\n%d", fnum); }</pre>		2								
3	<p>Convert the infix expression <math>((A+B)*C-(D-E))^{(F+G)}</math> to its equivalent Prefix expression by filling the structure given below (^ is exponentiation operator).</p> <table border="1"> <thead> <tr> <th>Scanned symbol</th><th>action taken (push, pop, add to prefix exprn)</th><th>Stack contents</th><th>Current Prefix exprn</th></tr> </thead> <tbody> <tr> <td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>	Scanned symbol	action taken (push, pop, add to prefix exprn)	Stack contents	Current Prefix exprn						3
Scanned symbol	action taken (push, pop, add to prefix exprn)	Stack contents	Current Prefix exprn								
4	<p>Write C functions to implement following operations of multiple stacks (number of stacks is 'n') using a single 1-D array having 'm' locations with following prototypes,</p> <p>i) <b>void push(int i, int item, STACK *S);</b> //push an item on i<sup>th</sup> stack</p> <p>ii) <b>int pop(int i, STACK *S);</b> //pop an item from i<sup>th</sup> stack</p> <p>Use the following <b>STACK</b> structure:</p>		2								

	<pre>typedef struct {     int stackArr[m];     int boundary[n+1];     int top[n+1]; }STACK;</pre> <p>Here, <i>boundary[i]</i> and <i>top[i]</i> represents boundary and top respectively for the <math>i^{\text{th}}</math> stack. Necessary validation check has to be done for stack overflow and underflow situations.</p>	
5	<p>Given a nonempty unsorted singly linked list with a list_pointer <b>first</b> pointing to the first node in the list. Write a function <b>void delete_multi(list_pointer *first)</b> which deletes multiple occurrences of a node, keeping only the first occurrence in the list without creating a new list. Also, do not sort the list. Use the following definition to represent each node in the list:</p> <pre>typedef struct list_node *list_pointer; struct list_node {     int data;     list_pointer link; };</pre> <p>Sample input:</p>  <p>Expected output for the above given input:</p> 	4
6	<p>Given a circular singly linked list with a list_pointer <b>first</b> pointing to the first node in the list, write a function <b>void reverse(list_pointer *first)</b> which reverses the given circular linked list by changing links between the nodes. Use the following definition to represent each node in the list:</p> <pre>typedef struct list_node *list_pointer; struct list_node {     int data;     list_pointer link; };</pre>	2
7	<p>Write a complete C program to do the following:</p> <p>(i) Define a function <b>Nodeptr CreateDLL(char str[])</b> which takes a string as parameter and creates a Doubly Linked List of characters and returns the pointer to the first node.</p> <p>(ii) Define a function <b>int IsPalindrome(Nodeptr first)</b> to check whether the string represented by the above doubly linked list pointed to by <b>first</b>, is a palindrome or not and return 1/0 accordingly. Do not use any additional data structure.</p> <p>Write a main function to read a string and create Doubly Linked of characters and check whether the string is a palindrome using above functions. Assume the following structure definition:</p> <pre>typedef struct NODE *Nodeptr; struct NODE{     char letter;     Nodeptr llink, rlink; };</pre>	4