RV UNIVERSITY
Go, change the world

School of Computer Science & Engineering
B.Tech(H) Program

Midterm : Model Question Paper
Academic Year: 2023-2024
Term: 29 Jan to 21 May 2024
Semester: II

Date: 16-4-2024
Time: 9.15 am to 10.25 am(70-Minutes)
Course Code: CS1020
Course Name: Data Structures & Algorithms
Max Marks: 25

USN:
Name:

S#	PART-A Question (MCQ)	Total Marks	CO	Bloom s Level *
1	Memory allocated during runtime is from _____	1	CO4	Under stand
2	Study the below piece of code: #include<stdio.h> #include<stdlib.h> #define arrsize 3 int main() { int myArray[10]; int *reallocarray = (int *)realloc(myArray, sizeof(int) * 2*arrsize); if (reallocarray == NULL) printf("\n \n %d is size memory chunk could not be allotted in heap by RE-ALLOC ",arrsize * 2); else printf("\n %d is size memory chunk allotted in heap by RE-ALLOC", arrsize * sizeof(int)); } The output is: **a)** 6 is size memory chunk could not be allotted in heap by RE-ALLOC **b)** 6 is size memory chunk allotted in heap by RE-ALLOC **c)** Program gives an "abort" runtime error	1	CO5	Under stand
3	In the original version of the Tower of Hanoi puzzle, as it was published in the 1890s by Edouard Lucas, a French mathematician, the world will end ´after 64 disks have been moved in this mystical tower. **a)** How many moves are made by the i^{th} largest disk ($1 \le i \le n$) in this Algorithm? **b)** How many moves does it take to move the largest disk from source to destination? **c)** How many moves does it take to create a valid stack of disks similar to what we finally need, but without the largest disk? **d)** How many moves does it take to complete the entire movement of disks	2 (0.5 x 4)	CO3	Under stand

	from source to destination?			
4	_____ is the number of years it will take if monks living in this mystical tower could move one disk per minute. (Assume that monks do not eat, sleep, or die.).	1	CO3	Analyze
	PART-B Question (Linked List Programming Question)			
5	Complete the code for a DLL below to print output as: 18 : Ahalya 28 : Fatima 38 : Mariam ********** `#include <stdio.h>` `#include <string.h>` `#include<stdlib.h>` `typedef struct student` `{` `  char name[10];` `  int age;` `  struct student *prev;` `  struct student *next;` `}node;` `int main()` `{` `  node *head, *newnode; //declare new pointers if necessary` `  newnode = (node *) malloc(3 * sizeof(node));` `  newnode->age = 18; strcpy(newnode->name, "Ahalya");` `  newnode->prev=NULL; newnode->next=NULL;` `  head = newnode; newnode++;` `  //write code to complete  if necessary` `  newnode->age = 28; strcpy(newnode->name, "Fatima");` `  // FIll the code for this node` `  newnode->age = 38; strcpy(newnode->name, "Mariam");` `  // FIll the code for this node` `newnode = head;` `  for(int i = 0; i<3; i++)` `  {` `  printf("\n %d : %s ", newnode->age, newnode->name); newnode++;` `  }` `}//end of main`	5	CO7	Apply
6	**a)** Rewrite the above program so that it becomes ***either*** a Singly Linked List ***or*** a Circular Double or Singly Linked List. **b)** Search for "Draupadi" in the list and print the result **c)** Insert "Sarah" before "Mariam" **d)** Delete "Mariam"	10	CO6 OR CO8	Apply

	e) Display list after these operations			
	PART-C Recursion Programming Question			
7	Below is a 'C' Code *using Iteration* for a particular program. Read and understand it. **a)** What does this program do? (1 Mark) ***b)*** *Write its Recursive Version (2.5 Marks)* **c)** Draw either the recursion call stack or recursion tree. ********** `#include <stdio.h>` `int Mystery(int array[], int x, int low, int high) {` `  // Repeat until the pointers low and high meet each other` `  while (low <= high) {` `  int mid = low + (high - low) / 2;` `  if (array[mid] == x)` `  return mid;` `  if (array[mid] < x)` `  low = mid + 1;` `  else` `  high = mid - 1;` `  }` `  return -1;` `}//end Mystery` `int main(void) {` `  int array[] = {3, 4, 5, 6, 7, 8, 9};` `  int n = sizeof(array) / sizeof(array[0]);` `  int x = 4;` `  int result = Mystery(array, x, 0, n - 1);` `  if (result == -1)` `  printf("Mystery Operation : Unsuccessful");` `  else` `  printf("Mystery Operation : Element is at index %d", result);` `  return 0;` `}//end main`	5	CO3	Apply