

**B.M.S. College of Engineering, Bengaluru-560019**

Autonomous Institute Affiliated to VTU

**January / February 2020 Semester End Make Up Examinations**Programme: **B.E.**Branch : **Computer Science & Engineering**Course Code: **19CS3PCDST**Course: **Data Structures**Semester : **III**Duration: **3 hrs.**Max Marks: **100**Date: **30.01.2020**

- Instructions:** 1. Answer any FIVE full questions, choosing one full question from each unit.  
2. Missing data, if any, may suitably assumed.

**UNIT - I**

- 1 a) What is dynamic memory allocation? Explain the functions used to dynamically allocate memory in C language. 05
- b) Analyze the given lines of code below and write the output with justification. 08
- ```
#include <stdio.h>
#define MAX 100
void Arrayfunc(int arr1[], int st, int last);
int main() {
    int arr1[MAX];
    int n, i;
    printf(" Input the number of elements to be stored in the array :");
    scanf("%d",&n);
    printf(" Input %d elements in the array :\n",n);
    for(i=0;i<n;i++) {
        scanf("%d",&arr1[i]);
    }
    Arrayfunc(arr1, (0+n)/2, n);
    return 0;
}

void Arrayfunc(int arr1[], int st, int last) {
    if(st >= last)
        return;

    printf("%d ", arr1[st]);
    Arrayfunc(arr1, st+1, last);
}
```
- c) Consider that Suresh has written a program that accepts the words and provides the reverse of all the words. To test this program he has given the words from the following sentence as the input: "Mom of Anna used to teach Malayalam and stats at a school in Gadag during noon." Write a program which Suresh had written to reverse each of the words in the sentence using stack and check whether the word is same as its reverse. 07

**Important Note:** Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.

**UNIT - II**

- 2 a) Explain Priority queue using a real time application. 04
- b) Consider a Queue of size 5 given below and perform the following operations in sequence. Represent the status of the Circular Queue and linear Queue after each operation. Also explain the difference between circular queue and linear queue with respect to the given queue. 10



- 1.Enqueue(34)
- 2.Dequeue()
- 3.Enqueue(30)
- 4.Dequeue()

- c) Consider a line of vehicles at a petrol refilling station. There are different types of vehicles waiting in line to fill their petrol tanks. Once the vehicle gets its petrol tank filled it leaves the petrol station. Suppose that the time taken by different vehicles in the line to fill the petrol tanks is in the following sequence: 30s, 840s, 50s, 90s, 300s and 500s. Develop a program to show how the line of vehicles gets formed when all vehicles arrive at the same time and also the way they leave the petrol station. 06

### UNIT - III

- 3 a) Differentiate between Linked list and array. 05
- b) Consider the following C code snippet wherein head pointer of the following singly linked list and a value of 4 is sent in the function call for func1. Analyze the code snippet and write the output for it along with the justification. 08
- singly linked list: 1->2->3->4->5->6->7->8
- n=4
- ```

struct node {
    int data;
    struct node *next;
};

struct node* func2(struct node* head, struct node* temp, int n){
    struct node *nextn=NULL, *cur=head, *prev=temp;
    int count=0;
    while(cur!=NULL && (count++)<n){
        nextn=cur->next;
        cur->next=prev;
        prev=cur;
        cur=nextn;
    }

    head=prev;
    return head;
}

void func1(struct node *head, int n) {
    struct node *temp;
    int count=0;
    temp=head;
    while((count++)<n){

```

```

        temp=temp->next;
    }

    head=func2(head,temp,n);
    display(head);
}

```

- c) A company is maintaining an inventory system which has the information about their products. The information such as product id and its cost for each product is maintained in the inventory as shown below. 07



Simulate the scenario by writing a C function wherein inventory manager should be able to sort the inventory list in ascending order based on the cost of the product.

**OR**

- 4 a) Explain the limitations of array implementation of linked list. 05  
 b) Analyze the following code snippet. Consider that head pointer for the following singly linked list is passed to the func function. 08

head: 26->78->67->12->90->1->34

k=4

Find the output of the code snippet and also explain the 'func' function's working.

```

struct node {
    int data;
    struct node *next;
};
typedef struct node * NODE;

```

```

NODE func(NODE head,int k) {
    int length=0,i;
    NODE p=head;

    while(p!=NULL)
    {
        p=p->next;
        length++;
    }
    if(length<k)
        return head;

    p=head;
    for(i=0;i<length-k+1;i++)
        p=p->next;
    return p;
}

```

- c) Harsha is browsing for an article on data structures through his browser in the internet. On the first web page he opened there was a link to another web page. 07

He clicked on that link and the browser displayed him another linked web page. He has done such N clicks which has taken him to new webpage every time. Now he wants to find out which webpage he was looking into after x clicks from his first webpage wherein  $x \leq N$ . Help him to know the webpage by simulating this scenario as a C function using a singly linked list. Write the appropriate insert function and the function to get the webpage after x clicks from first webpage.

#### UNIT - IV

- 5 a) Explain some of the applications of linked list. 05  
 b) Analyze the function given below, assume that head pointer to the following doubly linked list is passed to the function. Explain the output of the function for the following function call. 08

head: 1<->2<->3<->4<->5

Function call: func(head, 4)

```
struct node {
    int data;
    struct node *next;
    struct node *prev;
};
typedef struct node * NODE;
```

```
NODE func(NODE head , int n) {
    int count=1,i;
    NODE p,q,t;

    if(head==NULL)
        return head;
    p=head;
    while(p->next!=NULL) {
        count++;
        p=p->next;
    }

    if(n>count || n<=0)
        return head;
    q=head;
    for(i=1;i<n;i++) {
        q=q->next;
    }
    if(q->next!=NULL){
        t=q->next;
        p->next=head;
        head->prev=p;
        q->next=NULL;
        t->prev=NULL;
        head=t;
    }
    return head;
}
```

- c) Design an 11 entry hashtable for hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20 using the hash function  $(i+5) \bmod 11$ . Also show the collisions if they occur and resolve them using linear probing method. 07

**OR**

- 6 a) Explain the following terms: (i) Open Hashing and (ii) Closed Hashing. 05  
b) (i) Analyze the following Hashtable with hash function as  $K \% M$  where  $K$  is the key and  $M$  is the size of the hashtable. Identify the index which has the hash collisions which have been resolved using linear probing. 08

Index	Key
0	
1	
2	2
3	15
4	15
5	18
6	19
7	7
8	8
9	20
10	23
11	16
12	

- (ii) Consider the following function which inserts new node at the end of doubly Linked list which has two parameters: one is the head pointer to the list and the element to be inserted in the new node. Fill in the missing statements and complete the function.

```
struct node{
    int data;
    struct node * next;
    struct node * prev;
};
typedef struct node* NODE;
NODE insert(NODE head, int item)
{
    NODE p;
    NODE newn=(NODE)malloc(sizeof(struct node));
    newn->data=item;
    newn->prev=NULL;
```

```

newn->next=NULL;

if(head==NULL)
    return newn;
p=head;
while(-----)
{
    -----
}
-----
-----
return head;
}

```

- c) You are given with the pointers to the head nodes of two circular singly linked lists storing the following long positive numbers. Write a C function to perform the addition of these two long positive numbers using circular singly linked list assuming that two circular singly linked list have been created for the numbers. The input numbers have their digits stored in reverse order in the two circular singly linked list. 07
- Number 1: 123456009 is stored as 900654321  
 Number 2: 908743 is stored as 347809  
 Output:124364752

## UNIT - V

- 7 a) Explain the following with an example each: 05
- (i) Strictly or Fully Binary tree (ii) Almost complete Binary tree
- b) Analyze the following code snippet for the following binary search tree sequence 08
- Sequence of nodes in Binary Search Tree: 17, 4, 42, 0, 9, 23, 89  
 Function Call: value=func(root, 42);
- ```

struct node {
    int data;
    struct node *lchild;
    struct node *rchild;
};

int func(NODE root,int element) {
    int x=0;
    while(root!=NULL)
    {
        if(element<root->data){
            root=root->lchild;
            x++;
        }
        else if (element>root->data){
            root=root->rchild;
            x++;
        }
        else {
            return x;
        }
    }
    return -1;
}

```

}

Assuming that the insert function calls in the main function insert the nodes in the binary search in the sequence provided. Find out what is the function “func” doing? Justify your answer.

- c) Write a C function and function calls to insert the nodes of the binary search tree in such a way that we get the following traversals as postorder and inorder traversals of the tree. 07

Post order: 2,6,4,9,19,12,7

In order: 2,4,6,7,9,12,19

\*\*\*\*\*

B.M.S.C.E. - MAKEUP EXAM 2020