

Data Structures Objective evaluation Test 2 SE D

...

Hi SHINIT, when you submit this form, the owner will be able to see your name and email address.

* Required

1

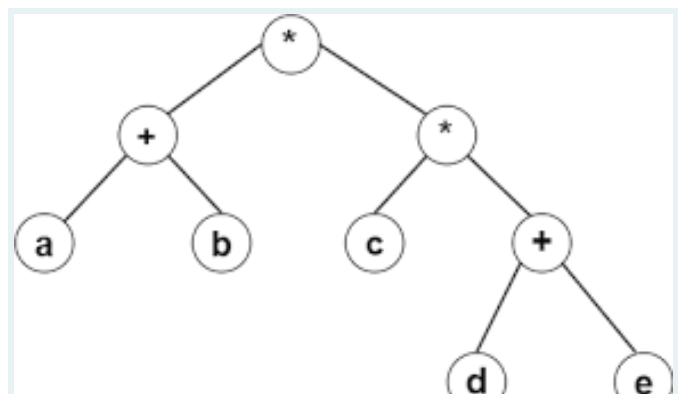
The preorder sequence of binary search tree is 30,22,17,27,24,38,34,48.

What is the post order sequence of the same tree? *

(1 Point)

- ☐ 17,22,24,27,30,34,38,48
- ☐ 30,22,38,17,27,34,48,24
- ☐ 17,22,27,24,34,48,38,30
- ☒ 17,24,27,22,34,48,38,30

2



The postorder traversal of the given expression tree is *
(1 Point)

- ☒ ab+de+c**
- ☐ a+b*c*d+e
- ☐ *+ab*c+de
- ☐ None

3

Algorithm DoSomething(Tree1, Tree2)

```
{  
    ans = false;  
    if(Tree1 = NULL AND Tree2 = NULL)  
        ans = True;  
    else if(Tree1->data == Tree2->data)  
    {  
        ans = Dosomething(LChild(Tree1), LChild(Tree2));  
        if(ans)  
            ans = Dosomething(RChild(Tree1) RChild(Tree2));  
    }  
    return ans;  
} *
```

(1 Point)

- ☐ a. Checking the leaf nodes
- ☒ b. Checking whether two trees are equal or not
- ☐ c. finding the left child
- ☐ d. finding the right child

4

What are the advantages of circular linked list over singly linked list? *
(1 Point)

- ☐ memory requirements are less
- ☐ The insert and delete operations can be implemented easily.
- ☐ The traversal can be done in forward and backward direction
- ☒ The traversal can be done from last node to first node

5

What is the operation performed by the given function -
void Listfun(struct node *head, int value)

```
{
    struct node *currP, *prevP;
    prevP = NULL;
    currP = head;
    while (currP != NULL && currP->element != value)
    {
        prevP = currP;
        currP = currP->next;
    }

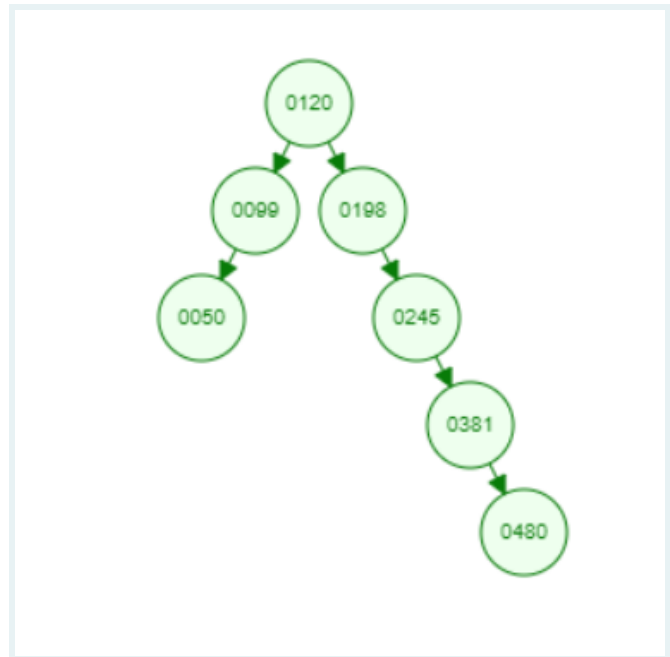
    if (currP->element == value) {
        if (prevP == NULL) {
            head = currP->next;
        } else {
            prevP->next = currP->next;
        }
        free(currP);
    } *
```

(1 Point)

- ☐ to delete the last node in the linked list

- ☒ to delete the node before the given value
- ☐ to delete the node with the given value
- ☐ to delete the node after the given value

6



A binary search tree stores values in the range 50 to 550. Consider the following sequence of keys:

1. 78,540,100,440,290,380,310
- 2.50,99,120,198, 245, 381, 480
3. 145, 250, 510,390, 350, 270,307
4. 550, 150,507,398,463,402, 270.

Which of the following statement is true ? *

(1 Point)

- ☐ a. 1,2 and 4 are inorder sequence of three different BST
- ☐ b. 1 is preorder sequence of some BST with 440 as root
- ☒ c. 2 is an inorder sequence of some BST where 120 is the root and 50 is leaf
- ☐ d. 4 is a postorder sequence of some BST with 150 as root

7

| Character | Code |
|-----------|---------|
| a | 1111110 |
| b | 1111111 |
| c | 111110 |
| d | 11110 |
| e | 1110 |
| f | 110 |
| g | 10 |
| h | 0 |

Consider the character code generated using Huffman coding as given in the table:

The encoded message: 111111011110100 is decoded to_____*

(1 Point)

- ☐ cbef
- ☐ bdgh
- ☒ adgh
- ☐ cdef

8

Roll No *

119A1076

9

What is the missing statement in the given code for inserting an element at the last position in circular linked list?

```
void insertNodeAtEnd(struct node *head, int val)
{
    struct node *p;
    struct node *temp=(struct node*)malloc(sizeof(struct node));
    temp->data=val;
    p=head;
    while(_____)
    {
        p=p->next;
    }
    _____;
    temp->next=head;
}
```

*

(1 Point)

- ☒ 1. p->next!=head 2. p->next=temp
- ☐ 1. p->next!=NULL 2. p->next=temp
- ☐ 1. p!=head 2. p->next=head
- ☐ 1. p->next!=head 2. p->next=head

10

Linked list is not suitable data structure for which of the following operations *

(1 Point)

- ☐ inserting a new element in the existing list
- ☐ deleting an element in the existing list
- ☒ sorting the elements stored in the list

 searching an element in the list

11

In following pseudocode to search for a given value in a linked list, identify the missing statement.

```
void search(struct node *start, int val)
```

```
{  
  struct node *ptr;  
  ptr = start;  
  while(_____  
  {  
    ptr = ptr -> next;  
  }  
  if (ptr->data==val)  
    printf ("data found");  
  else  
    printf("data not found");  
  } *
```

(1 Point)

- ☐ ptr->next!=NULL && ptr->data!=val
- ☒ ptr !=NULL && ptr->data!=val
- ☐ ptr->next!=NULL || ptr->data!=val
- ☐ ptr !=NULL || ptr->data!=val

Submit

This content is created by the owner of the form. The data you submit will be sent to the form owner. Microsoft is not responsible for the privacy or security practices of its customers, including those of this form owner. Never give out your password.

Powered by Microsoft Forms | [Privacy](#) and [cookies](#) | [Terms of use](#)

Powered by MICROSOFT FORMS | [Privacy and Cookies](#) | [Terms of Use](#)