Rajalakshmi Engineering College

Name: Yokitha selvakumar
Email: 241801327@rajalakshmi.edu.in
Roll no:
Phone: 6381952363
Branch: REC
Department: I AI & DS AF
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_MCQ_Updated

Attempt : 1
Total Mark : 20
Marks Obtained : 13

Section 1 : MCQ

1. What does the following code snippet do?

```
struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
newNode->data = value;
newNode->next = NULL;
newNode->prev = NULL;
```

Answer

Creates a new node and initializes its data to 'value'

Status : Correct ***Marks : 1/1***

2. How many pointers does a node in a doubly linked list have?

Answer

2

Status : Correct ***Marks : 1/1***

3. Which of the following information is stored in a doubly-linked list's nodes?

Answer

All of the mentioned options

Status : Correct ***Marks : 1/1***

4. Consider the following function that refers to the head of a Doubly Linked List as the parameter. Assume that a node of a doubly linked list has the previous pointer as prev and the next pointer as next.

Assume that the reference of the head of the following doubly linked list is passed to the below function 1 <--> 2 <--> 3 <--> 4 <--> 5 <-->6. What should be the modified linked list after the function call?

```
Procedure fun(head_ref: Pointer to Pointer of node)
  temp = NULL
  current = *head_ref

  While current is not NULL
    temp = current->prev
    current->prev = current->next
    current->next = temp
    current = current->prev
  End While

  If temp is not NULL
    *head_ref = temp->prev
  End If
End Procedure
```

Answer

6 &lt;--&gt; 5 &lt;--&gt; 4 &lt;--&gt; 3 &lt;--&gt; 1 &lt;--&gt; 2.

Status : Wrong ***Marks : 0/1***

5. What is a memory-efficient double-linked list?

Answer

Each node has only one pointer to traverse the list back and forth

Status : Wrong ***Marks : 0/1***

6. How do you delete a node from the middle of a doubly linked list?

Answer

All of the mentioned options

Status : Correct ***Marks : 1/1***

7. What is the main advantage of a two-way linked list over a one-way linked list?

Answer

Two-way linked lists allow for traversal in both directions.

Status : Correct ***Marks : 1/1***

8. Which of the following is true about the last node in a doubly linked list?

Answer

Its next pointer is NULL

Status : Correct ***Marks : 1/1***

9. What will be the output of the following code?

```
#include <stdio.h>
#include <stdlib.h>
```

```
struct Node {
   int data;
   struct Node* next;
   struct Node* prev;
};

int main() {
   struct Node* head = NULL;
   struct Node* temp = (struct Node*)malloc(sizeof(struct Node));
   temp->data = 2;
   temp->next = NULL;
   temp->prev = NULL;
   head = temp;
   printf("%d\n", head->data);
   free(temp);
   return 0;
}
```

Answer

2

Status : Correct ***Marks : 1/1***

10. Which code snippet correctly deletes a node with a given value from a doubly linked list?

```
void deleteNode(Node** head_ref, Node* del_node) {
   if (*head_ref == NULL || del_node == NULL) {
      return;
   }
   if (*head_ref == del_node) {
      *head_ref = del_node->next;
   }
   if (del_node->next != NULL) {
      del_node->next->prev = del_node->prev;
   }
   if (del_node->prev != NULL) {
      del_node->prev->next = del_node->next;
   }
```

```
    free(del_node);
}
```

Answer

Deletes the node at a given position in a doubly linked list.

Status : Wrong ***Marks : 0/1***

11. Which of the following is false about a doubly linked list?

Answer

Implementing a doubly linked list is easier than singly linked list

Status : Correct ***Marks : 1/1***

12. Where Fwd and Bwd represent forward and backward links to the adjacent elements of the list. Which of the following segments of code deletes the node pointed to by X from the doubly linked list, if it is assumed that X points to neither the first nor the last node of the list?

A doubly linked list is declared as

```
struct Node {
      int Value;
      struct Node *Fwd;
      struct Node *Bwd;
);
```

Answer

X.Bwd-&gt;Fwd = X.Bwd ; X-&gt;Fwd.Bwd = X.Bwd;

Status : Wrong ***Marks : 0/1***

13. What is the correct way to add a node at the beginning of a doubly linked list?

Answer

void addFirst(int data){   Node* newNode = new Node(data);   newNode-&gt;next = head;   if (head != NULL) {   head-&gt;prev =

newNode;   }   head = newNode;   }

Status : Correct ***Marks : 1/1***

14. Which of the following statements correctly creates a new node for a doubly linked list?

Answer

struct Node* newNode = (struct Node*) malloc(sizeof(struct Node));

Status : Correct ***Marks : 1/1***

15. What will be the effect of setting the prev pointer of a node to NULL in a doubly linked list?

Answer

The node will become the new head

Status : Correct ***Marks : 1/1***

16. What will be the output of the following program?

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* next;
   struct Node* prev;
};

int main() {
   struct Node* head = NULL;
   struct Node* tail = NULL;
   for (int i = 0; i < 5; i++) {
      struct Node* temp = (struct Node*)malloc(sizeof(struct Node));
      temp->data = i + 1;
      temp->prev = tail;
```

```
        temp->next = NULL;
        if (tail != NULL) {
            tail->next = temp;
        } else {
            head = temp;
        }
        tail = temp;
    }
    struct Node* current = head;
    while (current != NULL) {
        printf("%d ", current->data);
        current = current->next;
    }
    return 0;
}
```

Answer

1 2 3 4 5

Status : Correct ***Marks : 1/1***

17. Which pointer helps in traversing a doubly linked list in reverse order?

Answer

Status : Skipped ***Marks : 0/1***

18. Consider the provided pseudo code. How can you initialize an empty two-way linked list?

```
Define Structure Node
    data: Integer
    prev: Pointer to Node
    next: Pointer to Node
End Define

Define Structure TwoWayLinkedList
    head: Pointer to Node
    tail: Pointer to Node
End Define
```

Answer

Status : Skipped ***Marks : 0/1***

19. What happens if we insert a node at the beginning of a doubly linked list?

Answer

The previous pointer of the new node is NULL

Status : Correct ***Marks : 1/1***

20. How do you reverse a doubly linked list?

Answer

Status : Skipped ***Marks : 0/1***

Rajalakshmi Engineering College

Name: Yokitha selvakumar
Email: 241801327@rajalakshmi.edu.in
Roll no:
Phone: 6381952363
Branch: REC
Department: I AI & DS AF
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Your task is to create a program to manage a playlist of items. Each item is represented as a character, and you need to implement the following operations on the playlist.

Here are the main functionalities of the program:

Insert Item: The program should allow users to add items to the front and end of the playlist. Items are represented as characters.Display Playlist: The program should display the playlist containing the items that were added.

To implement this program, a doubly linked list data structure should be used, where each node contains an item character.

Input Format

The input consists of a sequence of space-separated characters, representing the items to be inserted into the doubly linked list.

The input is terminated by entering - (hyphen).

Output Format

The first line of output prints "Forward Playlist: " followed by the linked list after inserting the items at the end.

The second line prints "Backward Playlist: " followed by the linked list after inserting the items at the front.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: a b c -

Output: Forward Playlist: a b c
Backward Playlist: c b a

Answer

```
#include <stdio.h>
#include <stdlib.h>

struct Node {
    char item;
    struct Node* next;
    struct Node* prev;
};

// You are using GCC
void insertAtEnd(struct Node** head, char item) {
   struct Node*newNode=(struct Node*)malloc(sizeof(struct Node));
   newNode->item=item;
   newNode->next=NULL;
   newNode->prev=NULL;
   if(*head==NULL){
       *head=newNode;
       return;
   }
   struct Node*temp=*head;
   while(temp->next!=NULL){
      temp=temp->next;
   }
   temp->next=newNode;
   newNode->prev=temp;
}
void displayForward(struct Node* head) {
    struct Node*current=head;
    while(current!=NULL){
       printf("%c",current->item);
       current=current->next;
    }
    printf("\n");
}

void displayBackward(struct Node* tail) {
    if(tail==NULL)return;
    struct Node*current=tail;
    while(current!=NULL){
       printf("%c",current->item);
       current=current->prev;
    }
    printf("\n");
}

void freePlaylist(struct Node* head) {
    struct Node*current=head;
    while(current!=NULL){
       struct Node*temp=current;
       current=current->next;
       free(temp);
    }
}
int main() {
    struct Node* playlist = NULL;
    char item;

    while (1) {
       scanf(" %c", &item);
       if (item == '-') {
          break;
```

```
        }
        insertAtEnd(&playlist, item);
    }

    struct Node* tail = playlist;
    while (tail->next != NULL) {
        tail = tail->next;
    }

    printf("Forward Playlist: ");
    displayForward(playlist);

    printf("Backward Playlist: ");
    displayBackward(tail);

    freePlaylist(playlist);

    return 0;
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: Yokitha selvakumar
Email: 241801327@rajalakshmi.edu.in
Roll no:
Phone: 6381952363
Branch: REC
Department: I AI & DS AF
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_CY

Attempt : 1
Total Mark : 30
Marks Obtained : 30

Section 1 : Coding

1. Problem Statement

Imagine Anu is tasked with finding the middle element of a doubly linked list. Given a doubly linked list where each node contains an integer value and is inserted at the end, implement a program to find the middle element of the list. If the number of nodes is even, return the middle element pair.

Input Format

The first line of input consists of an integer N, representing the number of nodes in the doubly linked list.

The second line consists of N space-separated integers, representing the values of the nodes in the doubly linked list.

Output Format

The first line of output prints the space-separated elements of the doubly linked list.

The second line prints the middle element(s) of the doubly linked list, depending on whether the number of nodes is odd or even.

Refer to the sample outputs for the formatting specifications.

Sample Test Case

Input: 5
10 20 30 40 50

Output: 10 20 30 40 50
30

Answer

```
#include<stdio.h>
#include<stdlib.h>
struct node{
   int data;
   struct node*prev;
   struct node*next;
};
struct node*createnode(int data){
   struct node*newnode=(struct node*)malloc(sizeof(struct node));
   newnode->data=data;
   newnode->prev=NULL;
   newnode->next=NULL;
   return newnode;
}
void insertend(struct node**head,int data){
   struct node*newnode=createnode(data);
   if(*head==NULL){
      *head=newnode;
      return;
   }
   struct node*temp=*head;
   while(temp->next!=NULL)
   {
      temp=temp->next;
   }
   temp->next=newnode;
   newnode->prev=temp;
}
void printlist(struct node*head)
{
   struct node*temp=head;
   while(temp!=NULL){
      printf("%d",temp->data);
      temp=temp->next;
   }
   printf("\n");
}
void findmiddle(struct node*head,int n)
{
   struct node*slow=head;
   struct node*fast=head;
   int count=0;
   while(fast!=NULL&&fast->next!=NULL){
      slow=slow->next;
      fast=fast->next->next;
      count++;
   }
   if(n%2==0){
      printf("%d %d\n",slow->prev->data,slow->data);
   }else{
      printf("%d\n",slow->data);
   }
}
int main()
{
   int n,data;
   struct node*head=NULL;
   scanf("%d",&n);
   for(int i=0;i<n;i++)
   {
      scanf("%d",&data);
      insertend(&head,data);
   }
   printlist(head);
   findmiddle(head,n);
   return 0;
}
```

Rajalakshmi Engineering College

Name: Yokitha selvakumar
Email: 241801327@rajalakshmi.edu.in
Roll no:
Phone: 6381952363
Branch: REC
Department: I AI & DS AF
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_PAH

Attempt : 1
Total Mark : 50
Marks Obtained : 50

Section 1 : Coding

1. Problem Statement

Riya is developing a contact management system where recently added contacts should appear first. She decides to use a doubly linked list to store contact IDs in the order they are added. Initially, new contacts are inserted at the front of the list. However, sometimes she needs to insert a new contact at a specific position in the list based on priority.

Help Riya implement this system by performing the following operations:

Insert contact IDs at the front of the list as they are added.Insert a new contact at a given position in the list.

Input Format

The first line of input consists of an integer N, representing the initial size of the linked list.

The second line consists of N space-separated integers, representing the values of the linked list to be inserted at the front.

The third line consists of an integer position, representing the position at which the new value should be inserted (position starts from 1).

The fourth line consists of integer data, representing the new value to be inserted.

Output Format

The first line of output prints the original list after inserting initial elements to the front.

The second line prints the updated linked list after inserting the element at the specified position.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 4
10 20 30 40
3
25

Output: 40 30 20 10
40 30 25 20 10

Answer

```
#include<stdio.h>
#include<stdlib.h>
struct node{
   int data;
   node*prev;
   node*next;
};
struct node*createnode(int data){
   struct node*newnode=(struct node*)malloc(sizeof(struct node));
   newnode->data=data;
   newnode->next=NULL;
```

```
    newnode->prev=NULL;
    return newnode;
}
void insert(struct node**head,int data){
    struct node*newnode=createnode(data);
    if(*head==NULL){
        *head=newnode;
        return;
    }
    newnode->next=*head;
    (*head)->prev=newnode;
    *head=newnode;

}
void insertp(struct node**head,int newvalue,int p){
    if(p<1){
        printf("Invalid position");
        return;
    }
    struct node*newnode=createnode(newvalue);
    if(p==1){
        insert(head,newvalue);
        return;
    }
    struct node*temp=*head;
    int c=1;
    while(temp!=NULL&&c<p-1){
        temp=temp->next;
        c++;
    }
    if(temp==NULL){
        free(newnode);
        return;
    }
    newnode->next=temp->next;
    newnode->prev=temp;
    if(temp!=NULL){
        temp->next->prev=newnode;
    }
    temp->next=newnode;
}
void printlist(struct node* head){
    struct node* temp=head;
    while(temp!=NULL){
        printf("%d ",temp->data);
        temp=temp->next;
    }
    printf("\n");
}
int main(){
    int n,data,p,newvalue;
    struct node*head=NULL;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        scanf("%d",&data);
        insert(&head,data);
    }
    printlist(head);
    scanf("%d",&p);
    scanf("%d",&newvalue);
    insertp(&head,newvalue,p);
    printlist(head);
    return 0;
}
```

Status : Correct **Marks : 10/10**

2. Problem Statement

Bala is a student learning about the doubly linked list and its functionalities. He came across a problem where he wanted to create a doubly linked list by appending elements to the front of the list.

After populating the list, he wanted to delete the node at the given position from the beginning. Write a suitable code to help Bala.

Input Format

The first line contains an integer N, the number of elements in the doubly linked list.

The second line contains N integers separated by a space, the data values of the nodes in the doubly linked list.