Rajalakshmi Engineering College

Name: Prithika S
Email: 240701400@rajalakshmi.edu.in
Roll no: 240701400
Phone: 9790212894
Branch: REC
Department: I CSE FD
Batch: 2028
Degree: B.E - CSE

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;
  if(*head==NULL){
     newnode->prev= 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) {
    if(head==NULL)return;
    struct Node* temp=head;
    while(temp != NULL){
        printf("%c ",temp->item);
        temp=temp->next;
    }
    printf("\n");

}

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

void freePlaylist(struct Node* head) {
    struct Node* temp=head;
    while(head->next!=NULL){
        temp=head->next;
        free(head);
        head=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: Prithika S
Email: 240701400@rajalakshmi.edu.in
Roll no: 240701400
Phone: 9790212894
Branch: REC
Department: I CSE FD
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Moniksha, a chess coach organizing a tournament, needs a program to manage participant IDs efficiently. The program maintains a doubly linked list of IDs and offers two functions: Append to add IDs as students register, and Print Maximum ID to identify the highest ID for administrative tasks.

This tool streamlines tournament organization, allowing Moniksha to focus on coaching her students effectively.

Input Format

The first line consists of an integer n, representing the number of participant IDs to be added.

The second line consists of n space-separated integers representing the participant IDs.

Output Format

The output displays a single integer, representing the maximum participant ID.

If the list is empty, the output prints "Empty list!".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3
163 137 155

Output: 163

Answer

```
#include<stdio.h>
#include<stdlib.h>
typedef struct node{
   int data;
   struct node *prev,*next;
}node;
node* cnode(int val){
   node* newn=(node*)malloc(sizeof(node));
   newn->data=val;
   newn->prev=NULL;
   newn->next=NULL;
   return newn;
}
void insert(node** head,int val){
   node* newn=cnode(val);
   node*temp=*head;
   if(*head==NULL){
      *head=newn;
   }
   else{
      while(temp->next!=NULL){
         temp=temp->next;
      }
      temp->next=newn;
      newn->prev=temp;
```

```
    }
}
int larv(node* head){
    if(head==NULL){
        printf("Empty list!");

    }
    int max=head->data;
    node* temp=head->next;
    while(temp!=NULL)
    {
        if(temp->data>max)
            max=temp->data;
        temp=temp->next;
    }
    return max;
}
int main()
{
    node* head=NULL;
    int n,val;
    scanf("%d",&n);
    if(n==0){
        printf("Empty list!");
    }
    else{
    for(int i=0;i<n;i++){
        scanf("%d",&val);
        insert(&head,val);
    }
    printf("%d",larv(head));
    }
}
```

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

Rajalakshmi Engineering College

Name: Prithika S
Email: 240701400@rajalakshmi.edu.in
Roll no: 240701400
Phone: 9790212894
Branch: REC
Department: I CSE FD
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Bob is tasked with developing a company's employee record management system. The system needs to maintain a list of employee records using a doubly linked list. Each employee is represented by a unique integer ID.

Help Bob to complete a program that adds employee records at the front, traverses the list, and prints the same for each addition of employees to the list.

Input Format

The first line of input consists of an integer N, representing the number of employees.

The second line consists of N space-separated integers, representing the employee IDs.

Output Format

For each employee ID, the program prints "Node Inserted" followed by the current state of the doubly linked list in the next line, with the data values of each node separated by spaces.

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 4
101 102 103 104
Output: Node Inserted
101
Node Inserted
102 101
Node Inserted
103 102 101
Node Inserted
104 103 102 101
```

Answer

```
#include <iostream>
using namespace std;

struct node {
    int info;
    struct node* prev, * next;
};

struct node* start = NULL;

// You are using GCC
struct node* head=NULL;
void traverse() {
   //type your code here
   struct node* temp=head;
   printf("Node Inserted\n");
   while(temp){
       printf("%d",temp->info);
       temp=temp->next;
```

```
  }
  printf("\n");
}

void insertAtFront(int data) {
  //type your code here
  struct node* newn=(struct node*)malloc(sizeof(struct node));
  newn->info=data;
  newn->prev=NULL;
  newn->next=head;
  head=newn;
}

int main() {
    int n, data;
    cin >> n;
    for (int i = 0; i < n; ++i) {
        cin >> data;
        insertAtFront(data);
        traverse();
    }
    return 0;
}
```

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

Rajalakshmi Engineering College

Name: Prithika S
Email: 240701400@rajalakshmi.edu.in
Roll no: 240701400
Phone: 9790212894
Branch: REC
Department: I CSE FD
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

As part of a programming assignment in a data structures course, students are required to create a program to construct a singly linked list by inserting elements at the beginning.

You are an evaluator of the course and guide the students to complete the task.

Input Format

The first line of input consists of an integer N, which is the number of elements.

The second line consists of N space-separated integers.

Output Format

The output prints the singly linked list elements, after inserting them at the beginning.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
78 89 34 51 67

Output: 67 51 34 89 78

Answer

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

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

// You are using GCC
typedef struct Node node;
void insertAtFront(node** head,int x)
{
   node *newnode;
   newnode=(node *)malloc(sizeof(node));
   newnode->data=x;
   newnode->next=*head;
   *head=newnode;
}

void printList(node *head){
   Node *current=head;
   while(current!=NULL){
      printf("%d ",current->data);
      current=current->next;
   }
}

int main(){
   struct Node* head = NULL;
```

```
    int n;
    scanf("%d", &n);

    for (int i = 0; i < n; i++) {
        int activity;
        scanf("%d", &activity);
        insertAtFront(&head, activity);
    }

    printList(head);
    struct Node* current = head;
    while (current != NULL) {
        struct Node* temp = current;
        current = current->next;
        free(temp);
    }

    return 0;
}
```

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

Rajalakshmi Engineering College

Name: Prithika S
Email: 240701400@rajalakshmi.edu.in
Roll no: 240701400
Phone: 9790212894
Branch: REC
Department: I CSE FD
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Ashwin is tasked with developing a simple application to manage a list of items in a shop inventory using a doubly linked list. Each item in the inventory has a unique identification number. The application should allow users to perform the following operations:

Create a List of Items: Initialize the inventory with a given number of items. Each item will be assigned a unique number provided by the user and insert the elements at end of the list.

Delete an Item: Remove an item from the inventory at a specific position.

Display the Inventory: Show the list of items before and after deletion.

If the position provided for deletion is invalid (e.g., out of range), it should display an error message.

Input Format

The first line contains an integer n, representing the number of items to be initially entered into the inventory.

The second line contains n integers, each representing the unique identification number of an item separated by spaces.

The third line contains an integer p, representing the position of the item to be deleted from the inventory.

Output Format

The first line of output prints "Data entered in the list:" followed by the data values of each node in the doubly linked list before deletion.

If p is an invalid position, the output prints "Invalid position. Try again."

If p is a valid position, the output prints "After deletion the new list:" followed by the data values of each node in the doubly linked list after deletion.

Refer to the sample output for the formatting specifications.

Sample Test Case

```
Input: 4
1 2 3 4
5
Output: Data entered in the list:
 node 1 : 1
 node 2 : 2
 node 3 : 3
 node 4 : 4
Invalid position. Try again.
```

Answer

```
#include<stdio.h>
#include<stdlib.h>
typedef struct node{
    int data;
    struct node *prev,*next;
}node;
node* cnode(int val){
    node* newn=(node*)malloc(sizeof(node));
    newn->data=val;
    newn->prev=NULL;
    newn->next=NULL;
    return newn;
}
void insert(node** head,int data){
    node *newn=cnode(data);
    if(*head==NULL){
        *head=newn;
    }
    else{
        node* temp=*head;
        while(temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next=newn;
        newn->prev=temp;
    }
}
void dis(node* head){
    node* temp=head;
    printf("Data entered in the list:\n");
    for(int i=1; temp!=NULL; i++){
        printf("node %d : %d\n",i,temp->data);
        temp=temp->next;
    }
}
void del(node**head,int pos)
{
    node* temp=*head;
    node* trav=*head;
    int i=1;
    while(temp!=NULL&&i<pos){
        temp=temp->next;
        i++;
    }
    if(temp==NULL){
        printf("Invalid position. Try again.");
    }
    if(temp->prev==NULL){
        *head=temp->next;
        if(*head!=NULL)
            (*head)->prev=NULL;
    }
    else{
        temp->prev->next=temp->next;
        if(temp->next!=NULL)
            temp->next->prev=temp->prev;
    }
    free(temp);
    printf("After deletion the new list:\n");
    for(int i=1; trav!=NULL; i++){
        printf("node %d : %d\n",i,trav->data);
        trav=trav->next;
    }
}
int main()
{
    node* head=NULL;
    int n,val,pos;
    scanf("%d",&n);

    if(1<=n && n<=20) {
    for(int i=0; i<n; i++)
    {
        scanf("%d",&val);
        insert(&head,val);
    }
    dis(head);
    scanf("%d",&pos);
    if(1<=pos&&pos<=n){
        del(&head,pos);
    }
    else{
        printf("Invalid position. Try again.");
    }
    }
    else
    {
      printf("No data found in the list yet.Invalid position. try again.");
    }
}
```

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