Name: Sunmathi S

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 1

Attempt : 1 Total Mark : 10 Marks Obtained : 10

Section 1: Coding

1. Problem Statement

Janani is a tech enthusiast who loves working with polynomials. She wants to create a program that can add polynomial coefficients and provide the sum of their coefficients.

The polynomials will be represented as a linked list, where each node of the linked list contains a coefficient and an exponent. The polynomial is represented in the standard form with descending order of exponents.

Input Format

The first line of input consists of an integer n, representing the number of terms in the first polynomial.

The following n lines of input consist of two integers each: the coefficient and the exponent of the term in the first polynomial.

The next line of input consists of an integer m, representing the number of terms in the second polynomial.

The following m lines of input consist of two integers each: the coefficient and the exponent of the term in the second polynomial.

Output Format

The output prints the sum of the coefficients of the polynomials.

```
Input: 3
    22
    3 183
    40
    22
    31
    40
    Output: 18
    Answer
    // You are using GCC
    #include<stdio.h>
    #include<stdlib.h>
int c,e;
    struct Node{
      struct Node* next;
    };
    struct Node*add(struct Node*p,int c, int e){
      struct Node*t = (struct Node*)malloc(sizeof(struct Node));
      t->c=c;
      t->e=e;
      t->next=NULL;
     if(p==NULL){
        return t;
    struct Node*temp=p;
     while(temp->next!=NULI
```

```
24,180,1283
         temp=temp->next;
      temp->next=t;
      return p;
     int main(){
       int n,m,c,e,sum=0;
       struct Node *p1=NULL,*p2=NULL;
       struct Node*ptr1,*ptr2;
       scanf("%d",&n);
       for(int i=0;i< n;i++){
         (scanf("%d %d",&c,&e));
         p1=add(p1,c,e);
       scanf("%d",&m);
       for(int i=0;i<m;i++){
          (scanf("%d %d",&c,&e));
          p2 = add(p2,c,e);
       }
       ptr1=p1;
       ptr2=p2;
if(ptr1!=NULL && ptr1!=NULL && ptr1->e==ptr2->e){
    sum+=ptr1->c+r1
       while(ptr1!=NULL && ptr2!=NULL){
                                                         24,801283
            sum+=ptr1->c+ptr2->c;
            ptr2=ptr2->next;
          }else if (ptr1->e>ptr2->e){
            sum+=ptr1->c;
            ptr1=ptr1->next;
          }else{
            sum+=ptr2->c;
            ptr2=ptr2->next;
          }
       }
       while (ptr1!=NULL){
                                                         241801783
         sum+=ptr1->c;
          ptr1=ptr1->next;
```

24,801283

241801283

```
241801283
                                                      241801283
sum+=ptr2!=NULL
sum+=ptr2->c;
ptr2=ptr2->-
       while(ptr2!=NULL){
         ptr2=ptr2->next;
       printf("%d\n",sum);
       ptr1=p1;
       while(ptr2!=NULL){
         struct Node*temp=ptr2;
         ptr2=ptr2->next;
         free(temp);
       }
       return 0;
                          24,801283
Status : Correct
                                                                          Marks : 10/10
```

24,801283

241801283

24,80,783

241801283

241801283

24,80,1283

24,80,1283

Name: Sunmathi S

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1: Coding

1. Problem Statement

Arun is learning about data structures and algorithms. He needs your help in solving a specific problem related to a singly linked list.

Your task is to implement a program to delete a node at a given position. If the position is valid, the program should perform the deletion; otherwise, it should display an appropriate message.

Input Format

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

The second line consists of N space-separated elements of the linked list.

The third line consists of an integer x, representing the position to delete.

Position starts from 1.

Output Format

The output prints space-separated integers, representing the updated linked list after deleting the element at the given position.

If the position is not valid, print "Invalid position. Deletion not possible."

Refer to the sample output for formatting specifications.

```
Input: 5
82317
    Output: 8 3 1 7
    Answer
    #include <stdio.h>
    #include <stdlib.h>
    void insert(int);
   void display_List();
   void deleteNode(int);
   struct node {
      int data:
      struct node* next;
   } *head = NULL, *tail = NULL;
    void insert (int value){
      struct node*newnode=(struct node*)malloc(sizeof(struct node));
      newnode->data=value;
      newnode->next=NULL:
      if(head==NULL){
        head=newnode;
        tail=newnode;
      }else{
      tail->next=newnode;
        tail=newnode;
```

```
void display_List(){
      struct node* temp=head;
      while (temp!=NULL){
        printf("%d" ,temp->data);
        temp=temp->next;
      printf("\n");
    void deleteNode(int position){
      if(head==NULL||position<1){
        printf("Invalid position. Deletion not possible.");
return;}
    struct node* temp=head;
    if(position==1){
      head=head->next;
      free(temp);
      display_List();
      return;
    struct node* prev=NULL;
    for(int i=1;temp!=NULL&&i<position;i++){
      prev=temp;
      printf("Invalid position. Deletion not possible.");
return;
if(temp==NULL){
    prev->next=temp->next;
    free(temp);
    display_List();
    int main() {
                                                   241801783
      int num_elements, element, pos_to_delete;
scanf("%d", &num_elements);
```

```
for (int i = 0; i < num_elements; i++) {
    scanf("%d", &element);
    insert(element);
}

scanf("%d", &pos_to_delete);

deleteNode(pos_to_delete);

return 0;
}

Status: Correct

Marks: 10/10
```

241801283

241801283

24,80,1283

241801283

241801283

24,180,1283

24,80,1283

Name: Sunmathi S 1

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1: Coding

1. Problem Statement

Imagine you are working on a text processing tool and need to implement a feature that allows users to insert characters at a specific position.

Implement a program that takes user inputs to create a singly linked list of characters and inserts a new character after a given index in the list.

Input Format

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

The second line consists of a sequence of N characters, representing the linked list.

The third line consists of an integer index, representing the index(0-based) after

which the new character node needs to be inserted.

The fourth line consists of a character value representing the character to be inserted after the given index.

Output Format

If the provided index is out of bounds (larger than the list size):

- 1. The first line of output prints "Invalid index".
- 2. The second line prints "Updated list: " followed by the unchanged linked list values.

Otherwise, the output prints "Updated list: " followed by the updated linked list after inserting the new character after the given index.

Refer to the sample output for formatting specifications.

```
Input: 5
abcde
2
X ~8°
Output: Updated list: a b c X d e
Answer
#include<stdio.h>
#include<stdlib.h>
struct node{
  char data;
  struct node* next;
};
struct node* crenode(char data){
  struct node* newnode=(struct node*)malloc(sizeof(struct node));
  newnode->data=data;
  newnode->next=NULL;
 return newnode;
```

```
void append(struct node** head, char data){
   struct node* newnode=crenode(data);
      if(*head==NULL){
         *head=newnode;
      }else{
         struct node* temp=*head;
        while (temp->next!=NULL){
           temp=temp->next;
        temp->next=newnode;
      }
    }
    void printlist(struct node* head){
   struct node* temp=head
      while(temp!=NULL){
        printf("%c ",temp->data);
        temp=temp->next;
      printf("\n");
    int insafterinx(struct node* head,int index,char newchar){
      struct node* temp=head;
      int count=0;
      while(temp!=NULL){
        if(count==index){
       struct node* newnode=crenode(newchar);
          newnode->next=temp->next;
           temp->next=newnode;
           return 1;
        temp=temp->next;
        count++;
      return 0;
    int main()
אייייע ( %d",&n);
struct node* head=NULL;
for(int i=0;i<n;i++\f
      int n;
```

241801283

```
char data;
scanf(" °′
ar'
                                                                                   24,80,1283
                                                       24,801283
         ्रवाग(" %c",&data);
append(&head,data);
       int index;
       scanf("%d",&index);
       char newchar;
       scanf(" %c",&newchar);
       if(!insafterinx(head,index,newchar)){
          printf("Invalid index\n");
       printf("Updated list: ");
                                                       24,180,1283
return 0;
       printlist(head);
                                                                           Marks: 10/10
     Status: Correct
```

041801283

241801283

24,80,1283

241801283

241801283

24,180,1283

24,801283

Name: Sunmathi S 🕏

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



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.

```
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
     #include<stdio.h>
     #include<stdlib.h>
     // typedef struct Node{
     // int data;
     // struct Node* next;
struc
// }node;
     void insertAtFront(Node**head,int data){
       Node* newnode=(Node*)malloc(sizeof(Node));
       newnode->data=data:
       newnode->next=*head;
       *head=newnode;
     }
     void printList(Node*head){
ابر intf("%d",head->،
head=head->next;
         printf("%d",head->data);
```

```
241801283
         nic n,val;
Node*head=NULL;
scanf("%d",&n):
while(p=)*
     // int main()
     #{
     //
     //
            scanf("%d",&val);
     //
            insertfront(&head,val);
     //
     //
          display(head);
     //
     // }
struct Node* head = NULL;
int n:
       scanf("%d", &n);
       for (int i = 0; i < n; i++) {
          int activity;
          scanf("%d", &activity);
          insertAtFront(&head, activity);
       }
while (current != NULL) {
struct Node* temp
       printList(head);
                                                           24,180,1283
       struct Node* current = head;
          struct Node* temp = current;
          current = current->next;
          free(temp);
       }
       return 0;
     }
```

Status: Correct Marks: 10/10

241801783

24,180,1283

24,180,1283

241801283

24,80,1283

241801283

24,801283

Name: Sunmathi S 🕠

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 5

Attempt : 1 Total Mark : 10 Marks Obtained : 10

Section 1: Coding

1. Problem Statement

Imagine you are tasked with developing a simple GPA management system using a singly linked list. The system allows users to input student GPA values, insertion should happen at the front of the linked list, delete record by position, and display the updated list of student GPAs.

Input Format

The first line of input contains an integer n, representing the number of students.

The next n lines contain a single floating-point value representing the GPA of each student.

The last line contains an integer position, indicating the position at which a student record should be deleted. Position starts from 1.

Output Format

After deleting the data in the given position, display the output in the format "GPA: " followed by the GPA value, rounded off to one decimal place.

Refer to the sample output for formatting specifications.

```
Input: 4
    3.8
    3.2
    3.5
   4.1
    Output: GPA: 4.1
    GPA: 3.2
    GPA: 3.8
    Answer
    // You are using GCC
    #include<stdio.h>
    #include<stdlib.h>
    typedef struct Node
      float data;
      struct Node*next;
    }Node;
    void insertFront(Node**head,float data)
      Node*newnode=(Node*)malloc(sizeof(Node));
      newnode->data=data;
      newnode->next=*head;
      *head=newnode;
    void deleteAt(Node**head,int pos){
.Jue*temp
if(pos==1){
*head-
      Node*temp=*head;
        *head=temp->next;
```

```
241801283
   free(temp);
    return;
  for(int i=1;temp&&i<pos-1;i++)
  temp=temp->next;
  Node*del=temp->next;
  temp->next=del->next;
  free(del);
}
void display(Node*head){
  while(head)
    printf("GPA: %.1f\n",head->data);
                                                24,180,1283
                     241801283
   head=head->next;
int main()
  int n,pos;
  float val;
  Node*head=NULL;
  scanf("%d",&n);
  while(n--)
  {
    scanf("%f",&val);
    insertFront(&head,val);
                                                24,180,1783
    scanf("%d",&pos);
    deleteAt(&head,pos);
    display(head);
    return 0;
}
```

Status: Correct Marks: 10/10

241801783

241801283

241801283

24,80,1283

241801283

24,801283

Name: Sunmathi S

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 6

Attempt : 1 Total Mark : 10 Marks Obtained : 10

Section 1: Coding

1. Problem Statement

John is tasked with creating a program to manage student roll numbers using a singly linked list.

Write a program for John that accepts students' roll numbers, inserts them at the end of the linked list, and displays the numbers.

Input Format

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

The second line consists of N space-separated integers, representing the roll numbers of students.

Output Format

The output prints the space-separated integers singly linked list, after inserting the roll numbers of students at the end.

Refer to the sample output for formatting specifications.

```
Input: 5
    23 85 47 62 31
    Output: 23 85 47 62 31
   Answer
   // You are using GCC
#include<stdio.h>
   #include<stdlib.h>
   typedef struct Node
     int data;
      struct Node*next;
    }Node:
   void insertEnd(Node**head,int data)
     Node*newnode=(Node*)malloc(sizeof(Node));
     newnode->data=data;
if(*head==NULL)
     newnode->next=NULL;
        *head=newnode;
        return;
     Node*temp=*head;
     while(temp->next)
     temp=temp->next;
     temp->next=newnode;
   void display(Node*head)
      while(head)
        printf("%d",head->data)
```

```
head=head->next;
                                                                     24,180,1283
                                              241801283
                       24,801283
      int n,v;
      Node*head=NULL;
      scanf("%d",&n);
      while(n--)
        scanf("%d",&v);
        insertEnd(&head,v);
                                              241801283
                       24,801283
return 0;
      display(head);
                                                               Marks: 10/10
    Status: Correct
```

241801283

241801283

24,80,783

241801283

241801283

24,801283

241801283

Name: Sunmathi S

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 1_COD_Question 7

Attempt : 1 Total Mark : 10 Marks Obtained : 10

Section 1: Coding

1. Problem Statement

Dev is tasked with creating a program that efficiently finds the middle element of a linked list. The program should take user input to populate the linked list by inserting each element into the front of the list and then determining the middle element.

Assist Dev, as he needs to ensure that the middle element is accurately identified from the constructed singly linked list:

If it's an odd-length linked list, return the middle element. If it's an evenlength linked list, return the second middle element of the two elements.

Input Format

The first line of input consists of an integer n, representing the number of elements in the linked list.

The second line consists of n space-separated integers, representing the elements of the list.

Output Format

The first line of output displays the linked list after inserting elements at the front.

The second line displays "Middle Element: " followed by the middle element of the linked list.

Refer to the sample output for formatting specifications.

```
Input: 5
10 20 30 40 50
Output: 50 40 30 20 10
Middle Element: 30
Answer
#include <stdio.h>
#include <stdlib.h>
struct Node {
  int data;
struct Node* next;
// You are using GCC
Node* push(struct Node* head, int data){
  struct Node*newnode=(struct Node*)malloc(sizeof(struct Node));
  newnode->data=data;
  newnode->next=head:
   return newnode;
}
int printMiddle(struct Node*head){
  struct Node*slow=head,*fast=head;
```

```
while(fast!=NULL && fast->next!= NULL){
    slow=slow->next;
     fast=fast->next->next;
  return slow->data;
int main() {
  struct Node* head = NULL;
  int n;
  scanf("%d", &n);
                                                                             241801283
  int value;
 for (int i = 0; i < n; i++) {
     scanf("%d", &value);
    head = push(head, value);
  }
  struct Node* current = head;
  while (current != NULL) {
     printf("%d ", current->data);
     current = current->next;
  printf("\n");
int middle_element = printMiddle(head);
  printf("Middle Element: %d\n", middle_element);
  current = head;
  while (current != NULL) {
     struct Node* temp = current;
     current = current->next;
     free(temp);
                                                 241801783
  return 0;
```

Marks: 10/10

Status: Correct

Name: Sunmathi S

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



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.

```
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;
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;
```

```
24,180,1283
      while (temp->next != NULL){
        temp = temp->next;
      temp->next = newNode;
      newNode->prev = temp;
    }
    void displayForward(struct Node* head){
      struct Node* temp = head;
      while (temp != NULL){
         printf("%c ", temp->item);
         temp = temp->next;
      printf("\n");
    void displayBackward(struct Node* tail){
      struct Node* temp = tail
      while (temp != NULL){
         printf("%c ", temp->item);
         temp = temp->prev;
    }
      printf("\n");
    void freePlaylist(struct Node* head){
      struct Node* temp;
      while (head != NULL){
         temp = head;
        head = head->next;
         free(temp);
    int main() {
      struct Node* playlist = NULL;
      char item;
      while (1) {
         scanf(" %c", &item);
         if (item == '-') {
insertAtEnd(&playlist, item);
           break;
                                                    241801283
```

24,180,1283

```
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
```

041801283

241801283

24,180,1283

24,801283

241801283

24,180,1283

24,180,1283

Name: Sunmathi S

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



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.

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
   // You are using GCC
   #include <stdio.h>
   #include <stdlib.h>
   struct node {
      int id;
      struct node* prev;
      struct node* next;
   };
   struct node* head = NULL
   void append(int id) {
      struct node* newNode = (struct node*)malloc(sizeof(struct node));
      newNode->id = id;
      newNode->prev = NULL;
      newNode->next = NULL;
      if (head == NULL) {
        head = newNode;
      } else {
        struct node* temp = head;
        while (temp->next != NULL) {
          temp = temp->next;
```

```
24,801283
                                                     241801283
        temp->next = newNode;
        newNode->prev = temp;
     void printMaxID() {
       if (head == NULL) {
         printf("Empty list!\n");
         return;
       }
       int maxID = head->id;
       struct node* temp = head->next;
                                                                                241801283
     while (temp != NULL) {
         if (temp->id > maxID) {
           maxID = temp->id;
         temp = temp->next;
       printf("%d\n", maxID);
     int main() {
       int n, id;
                                                                                24,801283
                                                     24,180,1283
      for (int i = 0; i < n; i++) {
scanf("%d". &id):
         append(id);
       }
       printMaxID();
       return 0;
     }
     Status: Correct
                                                                         Marks: 10/10
24,180,1283
                          241801283
                                                     241801783
```

Name: Sunmathi S 1

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



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.

```
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;
    void traverse() {
      struct node* temp = start;
      while (temp != NULL) {
        printf("%d ", temp->info);
        temp = temp->next;
printf("\n");
```

```
void insertAtFront(int data) {
  struct node* newNode = (struct node*)malloc(sizeof(struct node));
   newNode->info = data;
  newNode->prev = NULL;
  newNode->next = start;
  if (start != NULL) {
     start->prev = newNode;
  }
  start = newNode;
  printf("Node Inserted\n");
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

24,180,1283

01283

24,801283

247801283

241801283

241801283

1801285

Name: Sunmathi S 🎷

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 4

Attempt : 1 Total Mark : 10 Marks Obtained : 10

Section 1: Coding

1. Problem Statement

Ravi is developing a student registration system for a college. To efficiently store and manage the student IDs, he decides to implement a doubly linked list where each node represents a student's ID.

In this system, each student's ID is stored sequentially, and the system needs to display all registered student IDs in the order they were entered.

Implement a program that creates a doubly linked list, inserts student IDs, and displays them in the same order.

Input Format

The first line contains an integer N the number of student IDs.

The second line contains N space-separated integers representing the student IDs.

Output Format

The output should display the single line containing N space-separated integers representing the student IDs stored in the doubly linked list.

Refer to the sample output for formatting specifications.

```
Sample Test Case
```

```
Input: 5
   10 20 30 40 50
Output: 10 20 30 40 50
   Answer
   // You are using GCC
   #include <stdio.h>
   #include <stdlib.h>
   struct node {
     int id;
     struct node* prev;
      struct node* next;
   struct node* head = NULI
   void append(int id) {
     struct node* newNode = (struct node*)malloc(sizeof(struct node));
     newNode->id = id:
     newNode->prev = NULL;
      newNode->next = NULL:
     if (head == NULL) {
        head = newNode:
     } else {
      struct node* temp = head;
        while (temp->next != NULL) {
          temp = temp->next;
```

```
241801283
                                                   24,801283
        temp->next = newNode;
        newNode->prev = temp;
    void display() {
      struct node* temp = head;
      while (temp != NULL) {
        printf("%d ", temp->id);
        temp = temp->next;
      }
      printf("\n");
                                                                             24,801283
                         241801283
int main() {
      int n, id;
      scanf("%d", &n);
      for (int i = 0; i < n; i++) {
        scanf("%d", &id);
        append(id);
      }
      display();
      return 0;
                         24,801283
                                                   241801283
Status : Correct
                                                                      Marks : 10/10
```

241801283

24,180,1283

24,180,1283

Name: Sunmathi S

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



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
```

// You are using GCC #include <stdio.h> #include <stdlib.h>

```
struct Node {
  int data;
  struct Node* prev;
  struct Node* next;
};
void append(struct Node** head, int value) {
  struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
  newNode->data = value;
  newNode->prev = NULL;
  newNode->next = NULL;
                                                                          241801283
  if (*head == NULL) {
   *head = newNode;
  } else {
    struct Node* temp = *head;
    while (temp->next != NULL)
       temp = temp->next;
    temp->next = newNode;
    newNode->prev = temp;
  }
}
void display(struct Node* head) {
  int pos = 1;
  struct Node* temp = head;
while (temp != NULL) {
    printf(" node %d : %d\n", pos, temp->data);
    temp = temp->next;
    pos++;
}
void deleteAtPosition(struct Node** head, int position) {
  struct Node* temp = *head;
  int count = 1;
  if (position == 1) {
    *head = temp->next;
    if (*head != NULL)
       (*head)->prev = NULL
```

```
241801283
    free(temp);
    return;
  while (temp != NULL && count < position) {
    temp = temp->next;
     count++;
  }
  if (temp == NULL)
    return;
  if (temp->prev != NULL)
                                                  24,801283
   temp->prev->next = temp->next;
of (temp->next != NULL)
     temp->next->prev = temp->prev;
  free(temp);
int main() {
  int n, p, val;
  scanf("%d", &n);
  struct Node* head = NULL;
                                                  241801783
  for (int i = 0; i < n; i++) {
    scanf("%d", &val);
    append(&head, val);
  scanf("%d", &p);
  printf("Data entered in the list:\n");
  display(head);
  if (p < 1 || p > n) {
    printf("Invalid position. Try again.\n");
  } else {
                                                  241801283
    deleteAtPosition(&head, p);
    printf("\n After deletion the new list:\n");
     display(head);
```

24,801283

241801283

24,180,1283

24,180,1283 24,80,1283 241801283 return 0; Marks: 10/10 Status: Correct 241801283 241801283

241801283

241801283

241801283

Name: Sunmathi S 1

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 3_COD_Question 1

Attempt : 1 Total Mark : 10 Marks Obtained : 10

Section 1: Coding

1. Problem Statement

In a coding competition, you are assigned a task to create a program that simulates a stack using a linked list.

The program should feature a menu-driven interface for pushing an integer to stack, popping, and displaying stack elements, with robust error handling for stack underflow situations. This challenge tests your data structure skills.

Input Format

The input consists of integers corresponding to the operation that needs to be performed:

Choice 1: Push the integer value onto the stack. If the choice is 1, the following input is a space-separated integer, representing the element to be pushed onto

the stack.

Choice 2: Pop the integer from the stack.

Choice 3: Display the elements in the stack.

Choice 4: Exit the program.

Output Format

The output displays messages according to the choice and the status of the stack:

If the choice is 1, push the given integer to the stack and display the following: "Pushed element: " followed by the value pushed.

If the choice is 2, pop the integer from the stack and display the following: "Popped element: " followed by the value popped.

If the choice is 2, and if the stack is empty without any elements, print "Stack is empty. Cannot pop."

If the choice is 3, print the elements in the stack: "Stack elements (top to bottom): " followed by the space-separated values.

If the choice is 3, and there are no elements in the stack, print "Stack is empty".

If the choice is 4, exit the program and display the following: "Exiting program".

If any other choice is entered, print "Invalid choice".

Refer to the sample input and output for the exact format.

Sample Test Case

```
Input: 13
    14
    3
    2
Output: Pushed element: 3
    Pushed element: 4
    Stack elements (top to bottom): 43
    Popped element: 4
    Stack elements (top to bottom): 3
    Exiting program
    Answer
    #include <stdio.h>
    #include <stdlib.h>
int data;
    struct Node {
      struct Node* next;
    struct Node* top = NULL;
    // You are using GCC
    void push(int value) {
      Node*newnode=(Node*)malloc(sizeof(Node));
      if(newnode==NULL){
        printf("Memory allocartion failed\n");
        return;
      newnode->data=value;
      newnode->next=top;
```

```
printf("Pushed element: %d\n",value);
    void pop() {
      if(top==NULL){
        printf("Stack is empty.Cannot pop.\n");
        return;
      }
      Node*temp=top;
      int poppedvalue=temp->data;
      top=top->next;
      free(temp);
    printf("Popped element: %d\n",poppedvalue);
    void displayStack() {
      if(top==NULL){
        printf("Stack is empty\n");
        return;
      }
      Node*current=top;
      printf("Stack elements(top to bottom): ");
      while(current!=NULL){
                                                    241801283
       printf("%d",current->data);
         current=current->next;
      printf("\n");
    }
    int main() {
      int choice, value;
      do {
         scanf("%d", &choice);
         switch (choice) {
           case 1:
                                                    24,180,1283
             scanf("%d", &value);
             push(value);
             break;
           case 2:
```

```
241801283
24,80,1283
                                                    241801283
             displayStack();
break;
se 4
           case 3:
           case 4:
             printf("Exiting program\n");
             return 0;
           default:
             printf("Invalid choice\n");
       } while (choice != 4);
                          24,801283
                                                    241801283
       return 0;
                                                                        Marks: 10/10
     Status: Correct
```

041801283

241801283

24,180,1283

241801283

241801283

24,180,1283

24,80,1283

Name: Sunmathi S 🕠

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 3_COD_Question 2

Attempt : 1 Total Mark : 10 Marks Obtained : 10

Section 1: Coding

1. Problem Statement

Sanjeev is in charge of managing a library's book storage, and he wants to create a program that simplifies this task. His goal is to implement a program that simulates a stack using an array.

Help him in writing a program that provides the following functionality:

Add Book ID to the Stack (Push): You can add a book ID to the top of the book stack. Remove Book ID from the Stack (Pop): You can remove the top book ID from the stack and display its details. If the stack is empty, you cannot remove any more book IDs.Display Books ID in the Stack (Display): You can view the books ID currently on the stack. Exit the Library: You can choose to exit the program.

Input Format

The input consists of integers corresponding to the operation that needs to be performed:

Choice 1: Push the book onto the stack. If the choice is 1, the following input is a space-separated integer, representing the ID of the book to be pushed onto the stack.

Choice 2: Pop the book ID from the stack.

Choice 3: Display the book ID in the stack.

Choice 4: Exit the program.

Output Format

The output displays messages according to the choice and the status of the stack:

- 1. If the choice is 1, push the given book ID to the stack and display the corresponding message.
- 2. If the choice is 2, pop the book ID from the stack and display the corresponding message.
- 3. If the choice is 2, and if the stack is empty without any book ID, print "Stack Underflow"
- 4. If the choice is 3, print the book IDs in the stack.
- 5. If the choice is 3, and there are book IDs in the stack, print "Stack is empty"
- 6. If the choice is 4, exit the program and display the corresponding message.
- 7. If any other choice is entered, print "Invalid choice"

Refer to the sample output for the exact text and format.

Sample Test Case

4

Output: Book ID 19 is pushed onto the stack Book ID 28 is pushed onto the stack

```
Book ID 28 is popped from the stack
    Book ID in the stack: 19
Book ID 19 is popped from the stack
    Exiting the program
    Answer
    #include <stdio.h>
    #define MAX 100 // Define maximum size of the stack
    // Declare the stack and top pointer
    int stack[MAX];
    int top = -1;
    // Function to push a book ID onto the stack
    void push(int bookID) {
      if (top == MAX - 1) {
        printf("Stack Overflow\n");
      } else {
        top++;
        stack[top] = bookID;
        printf("Book ID %d is pushed onto the stack\n", bookID);
      }
    }
    // Function to pop a book ID from the stack
    void pop() {
      if(top == -1) {
        printf("Stack Underflow\n");
      } else {
        printf("Book ID %d is popped from the stack\n", stack[top]);
        top--;
    }
    // Function to display book IDs in the stack
    void display() {
      if (top == -1) {
        printf("Stack is empty\n");
      } else {
        printf("Book ID in the stack:");
        for (int i = top; i >= 0; i--) {
           printf("%d ", stack[i]);
```

```
printf("\n");
// Main function to execute the program
int main() {
  int choice, bookID;
  while (1) {
    // Read the choice from user
    if (scanf("%d", &choice) != 1) break; // Exit if invalid input is given
    switch (choice) {
       case 1:
         // Push book ID onto the stack
         if (scanf("%d", &bookID) != 1) break; // Exit if invalid input for book ID
         push(bookID);
         break;
       case 2:
         // Pop book ID from the stack
         pop();
         break;
       case 3:
         // Display book IDs in the stack
         display();
         break;
       case 4:
         // Exit the program
         printf("Exiting the program\n");
         return 0;
       default:
         // Invalid choice
         printf("Invalid choice\n");
    }
  }
  return 0;
                                                                      Marks : 10/10
Status: Correct
```

Name: Sunmathi S

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 3_COD_Question 3

Attempt : 1 Total Mark : 10 Marks Obtained : 10

Section 1: Coding

1. Problem Statement

Sharon is developing a programming challenge for a coding competition.

The challenge revolves around implementing a character-based stack data structure using an array.

Sharon's project involves a stack that can perform the following operations:

Push a Character: Users can push a character onto the stack.Pop a Character: Users can pop a character from the stack, removing and displaying the top character.Display Stack: Users can view the current elements in the stack.Exit: Users can exit the stack operations application.

Write a program to help Sharon to implement a program that performs the given operations.

Input Format

The input consists of integers corresponding to the operation that needs to be performed:

Choice 1: Push the character onto the stack. If the choice is 1, the following input is a space-separated character, representing the character to be pushed onto the stack.

Choice 2: Pop the character from the stack.

Choice 3: Display the characters in the stack.

Choice 4: Exit the program.

Output Format

The output displays messages according to the choice and the status of the stack:

- 1. If the choice is 1, push the given character to the stack and display the pushed character having the prefix "Pushed: ".
- 2. If the choice is 2, undo the character from the stack and display the character that is popped having the prefix "Popped: ".
- 3. If the choice is 2, and if the stack is empty without any characters, print "Stack is empty. Nothing to pop."
- 4. If the choice is 3, print the elements in the stack having the prefix "Stack elements: ".
- 5. If the choice is 3, and there are no characters in the stack, print "Stack is empty."
- 6. If the choice is 4, exit the program.
- 7. If any other choice is entered, print "Invalid choice"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 2

4

Output: Stack is empty. Nothing to pop.

Answer

#include <stdio.h>

```
#include <stdbool.h>
#define MAX_SIZE 100
     char items[MAX_SIZE];
     int top = -1;
     void initialize() {
       top = -1;
     bool isFull() {
       return top == MAX_SIZE - 1;
     bool isEmpty() {
       return top == -1;
     void push(char c) {
       if(top == -1)
          initialize();
       if (top < MAX_SIZE - 1) {
          items[++top] = c;
          printf("Pushed: %c\n", c);
                                                        241801283
 void pop() {
       if (top == -1) {
          printf("Stack is empty. Nothing to pop.\n");
          printf("Popped: %c\n", items[top--]);
       }
     }
     void display() {
       if (top == -1) {
print
} else {
print
                                                        241801783
          printf("Stack is empty.\n");
          printf("Stack elements:");
          for (int i = top; i >= 0; i--) {
```

```
printf("%c ", items[i]);
rintf("\n");
   printf("\n");
}
     int main() {
        initialize();
        int choice;
        char value;
        while (true) {
          scanf("%d", &choice);
          switch (choice) {
             case 1:
               scanf(" %c", &value);
               push(value);
               break;
             case 2:
               pop();
               break;
             case 3:
               display();
               break;
             case 4:
return default:
prin+*
               return 0;
               printf("Invalid choice\n");
        return 0;
     }
```

Status: Correct Marks: 10/10

141801283

24180128?

24,180,1283

24,801283

Name: Sunmathi S

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 3_COD_Question 4

Attempt : 1 Total Mark : 10 Marks Obtained : 10

Section 1: Coding

1. Problem Statement

You are a software developer tasked with building a module for a scientific calculator application. The primary function of this module is to convert infix mathematical expressions, which are easier for users to read and write, into postfix notation (also known as Reverse Polish Notation). Postfix notation is more straightforward for the application to evaluate because it removes the need for parentheses and operator precedence rules.

The scientific calculator needs to handle various mathematical expressions with different operators and ensure the conversion is correct. Your task is to implement this infix-to-postfix conversion algorithm using a stack-based approach.

Example

Input:

a+b

Output:

ab+

Explanation:

The postfix representation of (a+b) is ab+.

Input Format

The input is a string, representing the infix expression.

Output Format

The output displays the postfix representation of the given infix expression.

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: a+(b*e)
Output: abe*+
Answer
```

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

struct Stack {
    int top;
    unsigned capacity;
    char* array;
};

struct Stack* createStack(unsigned capacity) {
    struct Stack* stack = (struct Stack*)malloc(sizeof(struct Stack));

if (!stack)
```

```
return NULL;
       stack->top = -1;
       stack->capacity = capacity;
       stack->array = (char*)malloc(stack->capacity * sizeof(char));
       return stack:
     }
     int isEmpty(struct Stack* stack) {
       return stack->top == -1;
     }
return stack->array[stack->top];
     char pop(struct Stack* stack) {
       if (!isEmpty(stack))
          return stack->array[stack->top--];
       return '$';
     }
     void push(struct Stack* stack, char op) {
       stack->array[++stack->top] = op;
                                                       241801783
     #include <ctype.h>
    void infixToPostfix(char* infix) {
       char postfix[100];
       int k = 0;
       int i;
       char stack[100];
       int top = -1;
       for (i = 0; infix[i] != '\0'; i++) {
          char ch = infix[i];
postfix[k++] = ch;
else if (ch
                                                       241801283
            stack[++top] = ch;
```

241801283

```
else if (ch == ')') {
             while (top != -1 && stack[top] != '(') {
               postfix[k++] = stack[top--];
             top--;
          }
          else {
             int prec = (ch == '^{\prime})? 3:
                    (ch == '*' || ch == '/')?2:
                    (ch == '+' || ch == '-') ? 1 : 0;
             int isRight = (ch == '^');
             while (top != -1) {
               char topOp = stack[top];
               int topPrec = (topOp == '^{\prime})? 3:
                        (topOp == '*' || topOp == '/') ? 2 :
                        (topOp == '+' || topOp == '-') ? 1 : 0;
               if ((prec < topPrec) || (prec == topPrec && !isRight)) {
                  postfix[k++] = stack[top--];
               } else {
                  break;
             stack[++top] = ch;
        while (top != -1) {
          postfix[k++] = stack[top--];
        }
        postfix[k] = '\0';
        printf("%s\n", postfix);
     }
     int main() {
exp[100];
scanf("%s", exp);
infor
        infixToPostfix(exp);
```

return 0; 24,180,1283 Marks : 10/10 Status: Correct

Name: Sunmathi S

Email: 241801283@rajalakshmi.edu.in

Roll no: 241801283 Phone: 9940554830

Branch: REC

Department: I AI & DS FD

Batch: 2028

Degree: B.E - AI & DS



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 3_COD_Question 5

Attempt : 1 Total Mark : 10 Marks Obtained : 10

Section 1: Coding

1. Problem Statement

Milton is a diligent clerk at a school who has been assigned the task of managing class schedules. The school has various sections, and Milton needs to keep track of the class schedules for each section using a stack-based system.

He uses a program that allows him to push, pop, and display class schedules for each section. Milton's program uses a stack data structure, and each class schedule is represented as a character. Help him write a program using a linked list.

Input Format

The input consists of integers corresponding to the operation that needs to be performed:

Choice 1: Push the character onto the stack. If the choice is 1, the following input is a space-separated character, representing the class schedule to be pushed onto the stack.

Choice 2: Pop class schedule from the stack

Choice 3: Display the class schedules in the stack.

Choice 4: Exit the program.

Output Format

The output displays messages according to the choice and the status of the stack:

- If the choice is 1, push the given class schedule to the stack and display the following: "Adding Section: [class schedule]"
- If the choice is 2, pop the class schedule from the stack and display the following: "Removing Section: [class schedule]"
- If the choice is 2, and if the stack is empty without any class schedules, print "Stack is empty. Cannot pop."
- If the choice is 3, print the class schedules in the stack in the following:
- "Enrolled Sections: " followed by the class schedules separated by space.
- If the choice is 3, and there are no class schedules in the stack, print "Stack is empty"
- If the choice is 4, exit the program and display the following: "Exiting the program"
 - If any other choice is entered, print "Invalid choice"

Refer to the sample output for the exact format.

Sample Test Case

Input: 1 d

3

```
Output: Adding Section: d
Adding Section: h
Enrolle
    Removing Section: h
    Enrolled Sections: d
    Exiting program
    Answer
    #include <stdio.h>
    #include <stdlib.h>
    struct Node {
   char data;
      struct Node* next;
    struct Node* top = NULL;
    void push(char value) {
      struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
      newNode->data = value;
      newNode->next = top;
      top = newNode;
      printf("Adding Section: %c\n", value);
                                                     241801283
    void pop() {
    if (top == NULL) {
         printf("Stack is empty. Cannot pop.\n");
      } else {
         char value = top->data;
         struct Node* temp = top;
         top = top->next;
         free(temp);
         printf("Removing Section: %c\n", value);
      }
    void displayStack() {
                                                     241801283
      if (top == NULL) {
print.
} else {
      printf("Stack is empty\n");
```

```
241801283
   printf("Enrolled Sections: ");
    struct Node* temp = top;
    while (temp != NULL) {
      printf("%c ", temp->data);
       temp = temp->next;
    }
    printf("\n");
  }
}
int main() {
  int choice;
  char value;
  do {
 scanf("%d", &choice);
    switch (choice) {
       case 1:
         scanf(" %c", &value);
         push(value);
         break;
       case 2:
         pop();
         break;
       case 3:
         displayStack();
         break;
                                                  24,180,1283
     case 4:
         printf("Exiting program\n");
         break;
       default:
         printf("Invalid choice\n");
  } while (choice != 4);
  return 0;
}
```

Status: Correct

24,180,1283

1783

24,80,1283

241801283

24,180,1283

Marks: 10/10