

Rajalakshmi Engineering College

Name: Nekhita Sri
Email: 241801184@rajalakshmi.edu.in
Roll no: 241801184
Phone: 8637459907
Branch: REC
Department: I AI & DS FC
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results



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: 1 3

1 4

3

2

3

4

Output: Pushed element: 3

Pushed element: 4

Stack elements (top to bottom): 4 3

Popped element: 4

Stack elements (top to bottom): 3

Exiting program

Answer

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

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

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

```
struct Node* top = NULL;
```

```
// You are using GCC
```

```
void push(int value) {
```

```
    //Type your code here
```

```
    printf("Pushed element: %d\n",value);
```

```
    struct Node*newnode=(struct Node*)malloc(sizeof(struct Node));
```

```
    newnode->data=value;
```

```
    if(top==NULL)
```

```
    {
```

```
        newnode->next=NULL;
```

```

        top=newnode;

        return;
    }
    else
    {
        newnode->next=top;
        top=newnode;
    }
}

```

```

void pop() {
    //Type your code here
    if(top==NULL)
    {
        printf("Stack is empty.Cannot pop.\n");
    }
    else
    {
        struct Node*temp=top;
        printf("Popped element: %d\n",top->data);
        top=top->next;
        free(temp);
        return;
    }
}

```

```

void displayStack() {
    //Type your code here
    struct Node*tail;
    tail=top;
    if(tail==NULL)
    {
        printf("Stack is empty\n");
    }
    else
    {
        printf("Stack elements (top to bottom): ");
        while(tail!=NULL)
        {
            printf("%d ",tail->data);
            tail=tail->next;
        }
    }
}

```

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

int main() {
    int choice, value;
    do {
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                scanf("%d", &value);
                push(value);
                break;
            case 2:
                pop();
                break;
            case 3:
                displayStack();
                break;
            case 4:
                printf("Exiting program\n");
                return 0;
            default:
                printf("Invalid choice\n");
        }
    } while (choice != 4);
    return 0;
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Nekhita Sri
Email: 241801184@rajalakshmi.edu.in
Roll no: 241801184
Phone: 8637459907
Branch: REC
Department: I AI & DS FC
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results



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

Input: 1 19

1 28

2

3

2

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

```
// You are using GCC
#include<stdio.h>
#define size 100
int stack[size];
int top=-1;
void push(int x)
{
    printf("Book ID %d is pushed onto the stack\n",x);
    if(top!=100)
    {
        top=top+1;
        stack[top]=x;
    }
}
void pop()
{
    if(top== -1)
    {
        printf("Stack Underflow\n");
    }
    else
    {
        printf("Book Id %d is popped from the stack\n",stack[top]);
        top=top-1;
    }
}
void disp()
{
    for(int i=top;i>=0;i--)
    {
        printf("%d ",stack[i]);
    }
    printf("\n");
}
int main()
{
```



```

int choice;
while(1)
{
    scanf("%d",&choice);
    if(choice==1)
    {
        int n;
        scanf("%d",&n);
        push(n);
    }
    else if(choice==2)
    {
        pop();
    }
    else if(choice==3)
    {
        if(top== -1)
        {
            printf("Stack is empty\n");
        }
        else
        {
            printf("Book ID in the stack: ");
            disp();
        }
    }
    else if(choice==4)
    {
        printf("Exiting the program\n");
        break;
    }
    else
    {
        printf("Invalid choice");
    }
}
}

```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Nekhita Sri
Email: 241801184@rajalakshmi.edu.in
Roll no: 241801184
Phone: 8637459907
Branch: REC
Department: I AI & DS FC
Batch: 2028
Degree: B.E - AI & DS

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() {
    printf("Node Inserted\n");
    struct node*pos=head;
    while(pos!=NULL)
    {
        printf("%d ",pos->info);
        pos=pos->next;
```

```

    }
    printf("\n");
    //type your code here
}

void insertAtFront(int data) {

//type your code here
    struct node*newnode=(struct node*)malloc(sizeof(struct node));
    newnode->info=data;
    newnode->prev=NULL;
    if(head==NULL)
    {
        newnode->prev=NULL;
        head=newnode;
    }
    else
    {
        newnode->next=head;
        head=newnode;
    }
}

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: Nekhita Sri
Email: 241801184@rajalakshmi.edu.in
Roll no: 241801184
Phone: 8637459907
Branch: REC
Department: I AI & DS FC
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results



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 data;
    struct node*next;
    struct node*prev;
};
typedef struct node node;
void insert(node**head,int n)
{
    node*newnode=(node*)malloc(sizeof(node));
    newnode->data=n;
    newnode->next=NULL;
    if(*head==NULL)
    {
        newnode->prev=NULL;
        *head=newnode;
        return;
    }
    else
    {
        node*pos=*head;
```

```

        while(pos->next!=NULL)
        {
            pos=pos->next;
        }
        newnode->prev=pos;
        pos->next=newnode;
    }
}
void disp(node*head)
{
    while(head!=NULL)
    {
        printf("%d ",head->data);
        head=head->next;
    }
}
int main()
{
    int n,num;
    scanf("%d",&n);
    node*head=NULL;
    for(int i=0;i<n;i++)
    {
        scanf("%d",&num);
        insert(&head,num);
    }
    disp(head);
}

```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Nekhita Sri
Email: 241801184@rajalakshmi.edu.in
Roll no: 241801184
Phone: 8637459907
Branch: REC
Department: I AI & DS FC
Batch: 2028
Degree: B.E - AI & DS

Scan to verify results



NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 2_COD_Question 5

Attempt : 2
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 d;
    struct node*p;
    struct node*n;
};
typedef struct node node;
void insert(node**h,int x)
{
    node*ne=(node*)malloc(sizeof(node));
    ne->d=x;
    ne->p=NULL;
    ne->n=NULL;
    if(*h==NULL)
    {
        *h=ne;
    }
    else
    {
        node*po=*h;
        while(po->n!=NULL)
        {
            po=po->n;
        }
        po->n=ne;
        ne->p=po;
    }
}

```

```

int deletep(node**h,int ps)
{
    if(*h==NULL)
    {
        return 0;
    }
    else
    {
        node*c=*h;
        int co=1;
        while(c!=NULL&&co<ps)
        {
            c=c->n;
            co++;
        }
    }
}

```

```

    }
    if(c==NULL)
    {
        return 0;
    }
    if(c==*h)
    {
        *h=c->n;
        if(*h!=NULL)
        {
            (*h)->p=NULL;
        }
    }
    else
    {
        if(c->p!=NULL)
        {
            c->p->n=c->n;
        }
        if(c->n!=NULL)
        {
            c->n->p=c->p;
        }
    }
    free(c);
    return 1;
}
void disp(node*h)
{
    printf("\n");
    node*pos=h;
    int i=1;
    while(pos!=NULL)
    {
        printf("node %d : %d ",i,pos->d);
        pos=pos->n;
        i++;
        printf("\n");
    }
}
int main()

```

```
{
    int n,p,items;
    node*h=NULL;
    scanf("%d",&n);
    printf("Data entered in the list: ");
    for(int i=0;i<n;i++)
    {
        scanf("%d",&items);
        insert(&h,items);
    }
    disp(h);
    scanf("%d",&p);
    printf("\n");
    if(deletep(&h,p))
    {
        printf("After deletion the new list: ");
        disp(h);
    }
    else
    {
        printf("Invalid position. Try again.");
    }
}
```

Status : Correct

Marks : 10/10