

**Name:- Shravani Navnath Bandavane**

**Class:- SY BSC(CS)**

**Div:-I1**

**Roll No:- SI1139**

**Subject:- GitHub project**

**Code:-**

**Q. Insert an element in a singly linked list at middle position.**

#include <stdio.h>

#include <stdlib.h>

// Node structure

struct Node {

int data;

struct Node\* next;

};

// Function to create a new node

struct Node\* createNode(int data) {

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

newNode->data = data;

newNode->next = NULL;

return newNode;

}

// Function to insert node at the end of the list

void insertAtEnd(struct Node\*\* head, int data) {

struct Node\* newNode = createNode(data);

if (\*head == NULL) {

\*head = newNode;

} else {

struct Node\* temp = \*head;

while (temp->next != NULL)

temp = temp->next;

temp->next = newNode;

}

}

// Function to insert node at a particular position

void insertAtPosition(struct Node\*\* head, int data, int position) {

struct Node\* newNode = createNode(data);

if (position == 1) {

newNode->next = \*head;

\*head = newNode;

return;

}

struct Node\* temp = \*head;

int i;

for (i = 1; i < position - 1 && temp != NULL; i++) {

temp = temp->next;

}

if (temp == NULL) {

printf("Invalid position!\n");

return;

}

newNode->next = temp->next;

temp->next = newNode;

}

// Function to display the linked list

void displayList(struct Node\* head) {

struct Node\* temp = head;

while (temp != NULL) {

printf("%d ", temp->data);

temp = temp->next;

}

printf("\n");

}

// Main function

int main() {

struct Node\* head = NULL;

int n, i, value, data, position;

printf("Enter number of nodes: ");

scanf("%d", &n);

printf("Enter %d node values:\n", n);

for (i = 0; i < n; i++) {

scanf("%d", &value);

insertAtEnd(&head, value);

}

printf("Enter data to insert: ");

scanf("%d", &data);

printf("Enter position to insert at: ");

scanf("%d", &position);

insertAtPosition(&head, data, position);

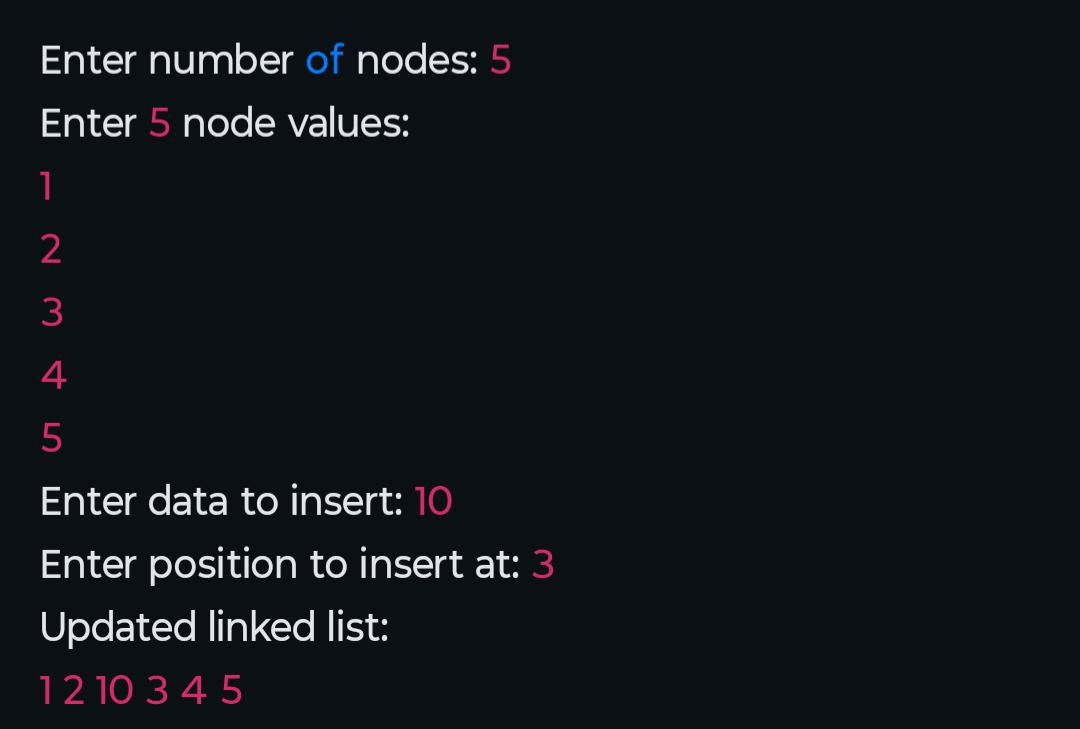
printf("Updated linked list:\n");

displayList(head);

return 0;

}

**Output**

****