Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering &Technology

Name :Dumpeta Varshitha .
Email :varshithadumpeta@gmail.com
Roll no:23071A1023
Phone :8919552327
Branch :VNRVJIET
Department :EIE
Batch :2027
Degree :B.Tech EIE

2023_27_II_EIE 1_Data Structures_lab

DATA STRUCTURES_CODING_WEEK 5

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : CODING

1. Problem Statement

Write a program to perform the following operations in a singly-linked list.

Insert the elements at the beginningInsert the elements at the endInsert the elements at the given position (Position starts from 1)

Insert the elements in the desired position and display the list after insertion.

Answer

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

#define MAX_SIZE 100 // Maximum size of the array

struct myNode {
   int val;
   int next;
};
```

```
int freeIndex = 0; // Next free index in the array

void insertAtEnd(struct myNode *list, int *head, int val) {
   int tmpIndex = freeIndex++;
   list[tmpIndex].val = val;
   list[tmpIndex].next = -1;
   if (*head == -1) {
      *head = tmpIndex;
   } else {
      int curr = *head;
      while (list[curr].next != -1) {
         curr = list[curr].next;
      }
      list[curr].next = tmpIndex;
   }
}

void insertAtBeginning(struct myNode *list, int *head, int val) {
   int tmpIndex = freeIndex++;
   list[tmpIndex].val = val;
   list[tmpIndex].next = *head;
   *head = tmpIndex;
}

void insertAtPos(struct myNode *list, int *head, int pos, int new_val) {
   int tmpIndex = freeIndex++;
   list[tmpIndex].val = new_val;
   list[tmpIndex].next = -1;
   int i = 1;
   int curr = *head;
   int last = -1;
   while (i < pos) {
      last = curr;
      curr = list[curr].next;
      i++;
   }
   if (last != -1) {
      list[last].next = tmpIndex;
   } else {
      *head = tmpIndex;
   }
```

```
    list[tmpIndex].next = curr;
}

void print(struct myNode *list, int head) {
    if (head == -1)
        return;
    int curr = head;
    printf("%d ", list[curr].val);
    print(list, list[curr].next);
}

int main() {
    int choice, new_val, pos;
    struct myNode myList[MAX_SIZE];
    int myListHead = -1;

    while (1) {
        scanf("%d", &choice);
        switch (choice) {
        case 1:
            scanf("%d", &new_val);
            insertAtBeginning(myList, &myListHead, new_val);
            break;
        case 2:
            scanf("%d", &new_val);
            insertAtEnd(myList, &myListHead, new_val);
            break;
        case 3:
            scanf("%d", &pos);
            scanf("%d", &new_val);
            insertAtPos(myList, &myListHead, pos, new_val);
            break;
        case 4:
            print(myList, myListHead);
            printf("\n");
            break;
        case 5:
            printf("Exiting");
            return 0;
        default:
            printf("Wrong choice\n");
            break;
        }
    }
}
```

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

2. Problem Statement

Write a program to reorder the given Linked list. Given a singly linked list: A0 A1 … An-1 An, reorder it to: A0 An A1 An-1 A2 An-2 …

For example: Given list is 1->2->3->4->5, the reordered list is 1->5->2->4->3.

Answer

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

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

struct Node* newNode(int key) {
    struct Node* temp = (struct Node*)malloc(sizeof(struct Node));
    temp->data = key;
    temp->next = NULL;
    return temp;
}

void reverselist(struct Node** head) {
    struct Node* prev = NULL, *curr = *head, *next;

    while (curr) {
        next = curr->next;
        curr->next = prev;
        prev = curr;
        curr = next;
    }
    *head = prev;
}

void printlist(struct Node* head) {
    while (head != NULL) {
        printf("%d ", head->data);
        if (head->next)
            printf("->");
        head = head->next;
    }
    printf("\n");
}

void rearrange(struct Node** head) {
    struct Node* slow = *head, *fast = slow->next;

    while (fast && fast->next) {
        slow = slow->next;
        fast = fast->next->next;
    }

    struct Node* head1 = *head;
    struct Node* head2 = slow->next;
    slow->next = NULL;

    reverselist(&head2);

    *head = newNode(0);

    struct Node* curr = *head;
    while (head1 || head2) {
        if (head1) {
            curr->next = head1;
            curr = curr->next;
            head1 = head1->next;
        }

        if (head2) {
            curr->next = head2;
            curr = curr->next;
            head2 = head2->next;
        }
    }

    *head = (*head)->next;
}

int main() {
    int n, data;
    scanf("%d%d", &n, &data);
    struct Node* head = newNode(data);
    struct Node* temp = head;

    for (int i = 0; i < n - 1; i++) {
        scanf("%d", &data);
        temp->next = newNode(data);
        temp = temp->next;
    }

    printlist(head);
    rearrange(&head);
    printlist(head);
    return 0;
}
```

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