#include <stdio.h>

#include <stdlib.h>

#define No\_Of\_Nodes 4

struct Node

{

  int item;

  struct Node\* next;

};

// insert at beginning

void insertBeginning(struct Node\*\* ref, int data)

{

  struct Node\* new\_node = (struct Node\*)malloc(sizeof(struct Node));

  new\_node->item = data;

  new\_node->next = \*ref;

  \*ref = new\_node;

}

void insertMiddle(struct Node\* node, int data)

{

  if (node == NULL)

  {

    printf("[!!] The previous node cannot be null");

    exit(0);

  }

  struct Node\* new\_node = (struct Node\*)malloc(sizeof(struct Node));

  new\_node->item = data;

  new\_node->next = node->next;

  node->next = new\_node;

}

// insert at end

void insertEnd(struct Node\*\* ref, int data)

{

  struct Node\* new\_node = (struct Node\*)malloc(sizeof(struct Node));

  struct Node\* last = \*ref;

  new\_node->item = data;

  new\_node->next = NULL;

  if (\*ref == NULL)

  {

    \*ref = new\_node;

    return;

  }

  while (last->next != NULL)

    last = last->next;

  last->next = new\_node;

  return;

}

// deletion of node

void deleteNode(struct Node\*\* ref, int key)

{

  struct Node \*temp = \*ref, \*prev;

  if (temp != NULL && temp->item == key)

  {

    \*ref = temp->next;

    free(temp);

    return;

  }

  while (temp != NULL && temp->item != key)

  {

    prev = temp;

    temp = temp->next;

  }

  if (temp == NULL) printf("[!!] Invalid Input, key is not present");

  prev->next = temp->next;

  free(temp);  // freeing the memory

}

// printing

void print(struct Node \*node)

{

  while (node != NULL)

  {

    printf(" %d -->", node->item);

    node = node->next;

  }

  printf(" NULL\n");

}

int main()

{

  struct Node\* head = NULL;  // empty

  int i, val, opt;

  char chk;

  // Creation

  printf("Creation Of Linked List => \n");

  printf("\n>> Enter the nodes which you want to insert in the linked list: \n");

  for(int i=0;i<No\_Of\_Nodes;i++)

  {

    printf(">> Enter data of node %d: ", i+1);

    fflush(stdin);

    scanf("%d", &val);

    insertBeginning(&head, val);

  }

  printf("\nCreated Linked List Successfully!!!\n");

  // Printing

  printf("\n>> Do you want to print the linked list now [y/n] : ");

  fflush(stdin);

  scanf("%c", &chk);

  if(chk == 'y')

  {

    printf("\nCreated Linked List => ");

    print(head);

  }

  else

  {

    printf("\nInsertion Of Nodes => ");

  }

  // insertion

  printf("\n>> Do you want to insert any node in the linked list [y/n] : ");

  fflush(stdin);

  scanf("%c", &chk);

  if(chk == 'y')

  {

    printf("\n>> Choose one of the following: ");

    printf("\n1: Beginning \t 2: Middle \t 3: End\n");

    fflush(stdin);

    scanf("%d", &opt);

    printf("\n>> Enter the data: ");

    scanf("%d", &val);

    switch(opt)

    {

      case 1:

        insertBeginning(&head, val);

        break;

      case 2:

        insertMiddle(head->next, val);

        break;

      case 3:

        insertEnd(&head, val);

        break;

      default:

        printf("[!!] Invalid input, Exitting...");

        exit(0);

    }

  }

  else

  {

    printf("\nNo nodes inserted!!!\n");

  }

  // printing

  printf("\n>> Do you want to print the linked list now [y/n] : ");

  fflush(stdin);

  scanf("%c", &chk);

  if(chk == 'y')

  {

    printf("\nCreated Linked List => ");

    print(head);

  }

  else

  {

    printf("\nDeletion Of Nodes => ");

  }

  // deletion

  printf("\n>> Do you want to delete any node of the linked list [y/n] :");

  fflush(stdin);

  scanf("%c", &chk);

  if(chk == 'y')

  {

    printf("\n>> Enter the data of the node which you want to delete: ");

    fflush(stdin);

    scanf("%d", &val);

    deleteNode(&head, val);

  }

  else

  {

    printf("\nNo nodes deleted!!!\n");

  }

  // printing

  printf("\nFinal Linked List => ");

  print(head);

  return 0;

}

Output

