Exp no. 5

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

typedef struct node {
   int data;
   struct node* next;
} node;

node* createlist();
node* insertbeg(node* head, int x);
node* insertend(node* head, int x);
node* insertmid(node* head, int x);
node* deletebeg(node* head);
node* deleteend(node* head);
node* deletemid(node* head);
void Printlist(node* head);

int main() {
   int choice, insertoption, deleteoption, x;
   node* head = NULL;
   printf("Welcome to the implementation of a singly linked list\n");

   do {
      printf("Please select an option from below:\n");
      printf("1. Create a List\n");
      printf("2. Insert a Node\n");
      printf("3. Delete a Node\n");
      printf("4. Print the Existing List\n");
      printf("5. Exit\n");
      printf("Enter your choice: ");
      scanf("%d", &choice);
      printf("\n");

      switch (choice) {
         case 1:
            head = createlist();
            break;
         case 2:
            do {
               printf("Select a position where you want to insert the new node:\n");
               printf("1. Beginning of the List\n");
               printf("2. At the End of the List\n");
               printf("3. Insert in Between\n");
               printf("4. Exit the Insert Operation\n");
               printf("Enter your choice: ");
               scanf("%d", &insertoption);

               switch (insertoption) {
                  case 1:
                     printf("Enter data to be inserted: ");
                     scanf("%d", &x);
                     head = insertbeg(head, x);
                     break;
```

```
            case 2:
                printf("Enter data to be inserted: ");
                scanf("%d", &x);
                head = insertend(head, x);
                break;
            case 3:
                printf("Enter data to be inserted: ");
                scanf("%d", &x);
                head = insertmid(head, x);
                break;
            case 4:
                printf("Insert operation exit\n");
                break;
            default:
                printf("Please enter a valid choice: 1, 2, 3, 4\n");
        }
    } while (insertoption != 4);
    printf("\n");
    break;
case 3:
    do {
        printf("Select a position from where you want to delete the element:\n");
        printf("1. Beginning of the List\n");
        printf("2. At the End of the List\n");
        printf("3. Somewhere in Between\n");
        printf("4. Exit the Delete Operation\n");
        printf("Enter your choice: ");
        scanf("%d", &deleteoption);

        switch (deleteoption) {
            case 1:
                head = deletebeg(head);
                break;
            case 2:
                head = deleteend(head);
                break;
            case 3:
                head = deletemid(head);
                break;
            case 4:
                printf("Delete Operation Exit\n");
                break;
            default:
                printf("Please enter a valid choice: 1, 2, 3, 4\n");
        }
    } while (deleteoption != 4);
    printf("\n");
    break;
case 4:
    Printlist(head);
    break;
case 5:
    printf("Exit: Program Finished!!\n");
    break;
default:
    printf("Please enter a valid choice: 1, 2, 3, 4, 5\n");
}
```

```
    } while (choice != 5);

    return 0;
}

node* createlist() {
    return NULL;
}

node* insertbeg(node* head, int x) {
    node* p = (node*)malloc(sizeof(node));
    p->data = x;
    p->next = head;
    head = p;
    return head;
}

node* insertend(node* head, int x) {
    node* p = (node*)malloc(sizeof(node));
    p->data = x;
    p->next = NULL;
    if (head == NULL)
        return p;
    node* q = head;
    while (q->next != NULL)
        q = q->next;
    q->next = p;
    return head;
}

node* insertmid(node* head, int x) {
    node* p = (node*)malloc(sizeof(node));
    p->data = x;
    p->next = NULL;
    printf("After which element do you want to insert the new element? Enter data: ");
    int y;
    scanf("%d", &y);
    node* q = head;
    while (q != NULL && q->data != y)
        q = q->next;
    if (q != NULL) {
        p->next = q->next;
        q->next = p;
    } else {
        printf("ERROR!! Data Not Found\n");
    }
    return head;
}

node* deletebeg(node* head) {
    if (head == NULL) {
        printf("Empty Linked List\n");
        return head;
    }
    node* p = head;
    head = head->next;
    free(p);
```

```
    return head;
}

node* deleteend(node* head) {
    if (head == NULL) {
        printf("Empty Linked List\n");
        return head;
    }
    node* p = head;
    if (head->next == NULL) {
        head = NULL;
        free(p);
        return head;
    }
    node* q = NULL;
    while (p->next != NULL) {
        q = p;
        p = p->next;
    }
    q->next = NULL;
    free(p);
    return head;
}

node* deletemid(node* head) {
    if (head == NULL) {
        printf("Empty Linked List\n");
        return head;
    }
    printf("Enter the data to be deleted: ");
    int x;
    scanf("%d", &x);
    if (head->data == x) {
        node* p = head;
        head = head->next;
        free(p);
        return head;
    }
    node* p = head;
    node* q = NULL;
    while (p != NULL && p->data != x) {
        q = p;
        p = p->next;
    }
    if (p == NULL) {
        printf("ERROR!! Data Not Found\n");
        return head;
    }
    q->next = p->next;
    free(p);
    return head;
}

void Printlist(node* head) {
    printf("[");
    node* p;
    for (p = head; p != NULL; p = p->next) {
```

```
        printf("%d\t", p->data);
    }
    printf("]\n\n");
}
```

output:

```
Enter your choice: 2

Select a position where you want to insert the new node:
1. Beginning of the List
2. At the End of the List
3. Insert in Between
4. Exit the Insert Operation
Enter your choice: 3
Enter data to be inserted: 8
After which element do you want to insert the new element? Enter data: 5
Select a position where you want to insert the new node:
1. Beginning of the List
2. At the End of the List
3. Insert in Between
4. Exit the Insert Operation
Enter your choice: 2
Enter data to be inserted: 10
Select a position where you want to insert the new node:
1. Beginning of the List
2. At the End of the List
3. Insert in Between
4. Exit the Insert Operation
Enter your choice: 4
Insert operation exit

Please select an option from below:
1. Create a List
2. Insert a Node
3. Delete a Node
4. Print the Existing List
5. Exit
Enter your choice: 4

[6      5       8       4       10      ]

Please select an option from below:
1. Create a List
2. Insert a Node
3. Delete a Node
4. Print the Existing List
5. Exit
Enter your choice:
```