

```

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

// Define node structure
struct node {
    int data;
    struct node *next;
};

struct node *head = NULL; // Initialize head pointer

void begininsert();
void lastinsert();
void randominsert();
void begin_delete();
void last_delete();
void random_delete();
void display();
void search();

int main() {
    int choice = 0;
    while (choice != 9) {
        printf("\n\n*****Main Menu*****\n");
        printf("\nChoose one option from the following list ...\n");
        printf("\n=====");
        printf("\n1. Insert in beginning\n2. Insert at last\n3. Insert at any random
location\n4. Delete from Beginning\n5. Delete from last\n6. Delete node after specified
location\n7. Search for an element\n8. Show\n9. Exit\n");
        printf("\nEnter your choice?\n");
        scanf("%d", &choice);

        switch (choice) {
            case 1: begininsert(); break;
            case 2: lastinsert(); break;
            case 3: randominsert(); break;
            case 4: begin_delete(); break;
            case 5: last_delete(); break;
            case 6: random_delete(); break;
            case 7: search(); break;
            case 8: display(); break;
            case 9: exit(0);
            default: printf("Please enter a valid choice..");
        }
    }
    return 0;
}

void begininsert() {
    struct node *ptr;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));

    if (ptr == NULL) {

```

```

        printf("\nOVERFLOW");
        return;
    }

    printf("\nEnter value: ");
    scanf("%d", &item);

    ptr->data = item;
    ptr->next = head;
    head = ptr;
    printf("\nNode inserted");
}

void lastinsert() {
    struct node *ptr, *temp;
    int item;

    ptr = (struct node*) malloc(sizeof(struct node));
    if (ptr == NULL) {
        printf("\nOVERFLOW");
        return;
    }

    printf("\nEnter value: ");
    scanf("%d", &item);

    ptr->data = item;
    ptr->next = NULL;

    if (head == NULL) {
        head = ptr;
        printf("\nNode inserted");
        return;
    }

    temp = head;
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->next = ptr;
    printf("\nNode inserted");
}

void randominsert() {
    struct node *ptr, *temp;
    int item, loc, i;

    ptr = (struct node *)malloc(sizeof(struct node));
    if (ptr == NULL) {
        printf("\nOVERFLOW");
        return;
    }

    printf("\nEnter value: ");

```

```

scanf("%d", &item);

printf("\nEnter location after which to insert: ");
scanf("%d", &loc);

temp = head;
for (i = 0; i < loc - 1; i++) {
    if (temp == NULL) {
        printf("\nCan't insert\n");
        return;
    }
    temp = temp->next;
}

ptr->data = item;
ptr->next = temp->next;
temp->next = ptr;
printf("\nNode inserted");
}

void begin_delete() {
    struct node *ptr;

    if (head == NULL) {
        printf("\nList is empty\n");
        return;
    }

    ptr = head;
    head = head->next;
    free(ptr);

    printf("\nNode deleted from beginning\n");
}

void last_delete() {
    struct node *ptr, *ptr1;

    if (head == NULL) {
        printf("\nList is empty\n");
        return;
    }

    if (head->next == NULL) {
        free(head);
        head = NULL;
        printf("\nOnly node deleted\n");
        return;
    }

    ptr = head;
    while (ptr->next->next != NULL) {
        ptr = ptr->next;
    }

```

```

    ptr1 = ptr->next;
    ptr->next = NULL;
    free(ptr1);

    printf("\nDeleted node from last\n");
}

void random_delete() {
    struct node *ptr, *ptr1;
    int loc, i;

    printf("\nEnter location after which to delete: ");
    scanf("%d", &loc);

    ptr = head;
    for (i = 0; i < loc - 1; i++) {
        if (ptr == NULL) {
            printf("\nCan't delete\n");
            return;
        }
        ptr = ptr->next;
    }

    if (ptr == NULL || ptr->next == NULL) {
        printf("\nCan't delete\n");
        return;
    }

    ptr1 = ptr->next;
    ptr->next = ptr1->next;
    free(ptr1);

    printf("\nDeleted node %d", loc + 1);
}

void search() {
    struct node *ptr;
    int item, i = 1, flag = 0;

    ptr = head;
    if (ptr == NULL) {
        printf("\nEmpty List\n");
        return;
    }

    printf("\nEnter item to search: ");
    scanf("%d", &item);

    while (ptr != NULL) {
        if (ptr->data == item) {
            printf("Item found at location %d\n", i);
            flag = 1;
            break;
        }
    }
}

```

```

        }
        ptr = ptr->next;
        i++;
    }

    if (flag == 0) {
        printf("Item not found\n");
    }
}

void display() {
    struct node *ptr;
    ptr = head;

    if (ptr == NULL) {
        printf("\nNothing to print\n");
        return;
    }

    printf("\nPrinting values:\n");
    while (ptr != NULL) {
        printf("%d -> ", ptr->data);
        ptr = ptr->next;
    }
    printf("NULL\n");
}

```