```
#include <iostream>
using namespace std;

class node {
public:
   int data;
   node* next;

   node(int d1, node* n1) {
      data = d1;
      next = n1;
   }

   node(int d1) {
      data = d1;
      next = nullptr;
   }
};

node* create(node* head) {
   int value = 0;
   node* y = head;

   while (true) {
      cout << "Enter number (enter -1 to stop): ";
      cin >> value;
```

```
        if (value == -1)
            break;

        node* temp = new node(value, nullptr);
        y->next = temp;
        y = temp;
    }

    return head;
}

node* insertbeg(node* head) {
    int val;
    cout << "Enter element to insert at the beginning: ";
    cin >> val;
    node* newhead = new node(val, head);
    return newhead;
}

node* insertend(node* head) {
    int val;
    cout << "Enter element to insert at the end: ";
    cin >> val;

    node* temp = head;
```

```
    while (temp->next != nullptr) {
        temp = temp->next;
    }

    temp->next = new node(val, nullptr);
    return head;
}

node* insertk(node* head, int k) {
    int x;
    cout << "Enter element to be inserted at position " << k << ": ";
    cin >> x;

    if (k == 1) {
        node* ele = new node(x, head);
        return ele;
    }

    int c = 1;
    node* temp = head;

    while (temp != nullptr) {
        if (c == k - 1) {
            node* ele = new node(x, temp->next);
            temp->next = ele;
            break;
```

```
        }
        c++;
        temp = temp->next;
    }

    return head;
}

node* deletehead(node* head) {
    if (head == nullptr)
        return nullptr;

    node* temp = head;
    head = head->next;
    delete temp;
    return head;
}

node* deletend(node* head) {
    if (head == nullptr || head->next == nullptr) {
        delete head;
        return nullptr;
    }

    node* temp = head;
    while (temp->next->next != nullptr) {
```

```
        temp = temp->next;
    }

    delete temp->next;
    temp->next = nullptr;
    return head;
}

node* deletek(node* head, int k) {
    if (k == 1 && head != nullptr) {
        node* temp = head;
        head = head->next;
        delete temp;
        return head;
    }

    int c = 1;
    node* temp = head;
    node* prev = nullptr;

    while (temp != nullptr) {
        if (c == k) {
            if (prev != nullptr) {
                prev->next = temp->next;
                delete temp;
            }
```

```
            break;
        }

        prev = temp;
        temp = temp->next;
        c++;
    }

    return head;
}

void display(node* head) {
    node* temp = head;
    while (temp != nullptr) {
        cout << temp->data << " ";
        temp = temp->next;
    }
    cout << endl;
}

int search(node* head, int element) {
    node* temp = head;
    while (temp != nullptr) {
        if (temp->data == element)
            return 1;
        temp = temp->next;
```

```
    }
    return 0;
}

int main() {
    node* head = nullptr;
    int op;
    char a='y';
    while(a=='y')
    {
        cout << "1. Create\n2. Insertion\n3. Deletion\n4. Search\n";
        cout << "Enter option: ";
        cin >> op;
        switch (op) {
            case 1:{
                int d1;
                cout << "Enter first element: ";
                cin >> d1;
                head = new node(d1);
                head = create(head);
                display(head);
                break;

            }
        case 2: {
            int op1;
```

```
    cout << "1. Insertion at beginning\n2. Insertion at end\n3. Insertion at position\n";
    cout << "Enter option: ";
    cin >> op1;
    switch (op1) {
      case 1:
        head = insertbeg(head);
        break;
      case 2:
        head = insertend(head);
        break;
      case 3: {
        int k;
        cout << "Enter position: ";
        cin >> k;
        head = insertk(head, k);
        break;
      }
    }
    display(head);
    break;
  }
  case 3: {
    int op2;
    cout << "1. Deletion of head\n2. Deletion of tail\n3. Deletion at position\n";
    cout << "Enter option: ";
    cin >> op2;
```

```
    switch (op2) {
        case 1:
            head = deletehead(head);
            break;
        case 2:
            head = deletend(head);
            break;
        case 3: {
            int k;
            cout << "Enter position: ";
            cin >> k;
            head = deletek(head, k);
            break;
        }
    }
    display(head);
    break;
}
case 4: {
    int element;
    cout << "Enter element to be searched: ";
    cin >> element;
    int flag = search(head, element);
    if (flag)
        cout << "Element found!\n";
    else
```

```
            cout << "Element not found!\n";
            break;
        }
        default:
            cout << "Invalid option!\n";
    }
    printf("do you want to continue (y/n):");
    scanf(" %c",&a);


    }







    return 0;
}
```