# Source Code

```
#include <iostream>

struct Node
{
    float m_data;
    Node* m_next;
    Node(float data = 0) : m_data(data), m_next(nullptr) {}
};

class singlyLinkedList
{
public:
    singlyLinkedList() : m_head(nullptr) {}

    void insert_beg(float data)
    {
        Node* new_node = new Node(data);
        new_node->m_next = m_head;
        m_head = new_node;
    }

    void insert_end(float data)
    {
        Node* new_node = new Node(data);
        if (!m_head)
        {
            m_head = new_node;
            return;
        }
        Node* current = m_head;
        while (current->m_next) current = current->m_next;
        current->m_next = new_node;
    }

    void insert_spec(float data, unsigned position)
    {
        Node* new_node = new Node(data);
        if (position == 0)
        {
            new_node->m_next = m_head;
            m_head = new_node;
            return;
        }

        Node* current = m_head;
        for (unsigned i = 1; i < position && current; ++i) current = current->m_next;
        if (!current)
        {
            std::cerr << "Invalid position.\n";
            delete new_node;
            return;
        }
        new_node->m_next = current->m_next;
        current->m_next = new_node;
    }

    void delete_beg()
    {
        if (m_head)
        {
            Node* temp = m_head;
            m_head = m_head->m_next;
            delete temp;
            return;
        }
        std::cerr << "List is empty.\n";
    }

    void delete_end()
    {
        if (!m_head)
        {
            std::cerr << "List is empty.\n";
            return;
        }
        if (!m_head->m_next)
        {
            delete m_head;
            m_head = nullptr;
            return;
        }
        Node* current = m_head;
        while (current->m_next->m_next) current = current->m_next;
        delete current->m_next;
        current->m_next = nullptr;
    }

    void delete_spec(unsigned position)
    {
        if (position == 0)
        {
            delete_beg();
            return;
        }
        Node* current = m_head;
        for (unsigned i = 1; i < position && current; ++i) current = current->m_next;
        if (!current || !current->m_next)
        {
            std::cerr << "Invalid position." << std::endl;
            return;
        }
        Node* temp = current->m_next;
        current->m_next = current->m_next->m_next;
        delete temp;
    }

    void display()
    {
        Node* current = m_head;
        if (!current)
        {
            std::cout << "List is empty.\n";
            return;
        }
        std::cout << "The data in the list is: \n";
        while (current)
        {
            std::cout << current->m_data << " ";
            current = current->m_next;
        }
        std::cout << "\n";
    }

    void search(float data)
    {
        unsigned position = 0;
        Node *current = m_head;
        while (current)
        {
            if(current->m_data == data)
            {
                std::cout<<"Data was found at position "<<position<<"\n";
                return;
            }
            position++;
            current = current->m_next;
        }
        std::cout<<"Data is not in the list.\n";
    }

    ~singlyLinkedList()
    {
        Node* current = m_head;
        while (current)
        {
            Node* next = current->m_next;
            delete current;
            current = next;
        }
    }

private:
    Node* m_head;
};
```

```
int main()
{
    singlyLinkedList list;
    float data;
    unsigned position;

    while(1)
    {
        unsigned choice = 0;
        std::cout << "\nDisplay Menu\n"
                  << "(0) to insert at beginning,\n"
                  << "(1) to insert at end,\n"
                  << "(2) to insert at position,\n"
                  << "(3) to delete from the beginning,\n"
                  << "(4) to delete from the end,\n"
                  << "(5) to delete from a position,\n"
                  << "(6) to display the list,\n"
                  << "(7) to search for data in the list\n"
                  << "(8) to exit the program\n";

        std::cin >> choice;

        switch(choice)
        {
            case 0:
                std::cout << "\nEnter the data to insert at the beginning.\n";
                std::cin >> data;
                list.insert_beg(data);
                break;
            case 1:
                std::cout << "\nEnter the data to insert at the end.\n";
                std::cin >> data;
                list.insert_end(data);
                break;
            case 2:
                std::cout << "Enter the position to insert data [0-n]\n";
                std::cin >> position;
                std::cout << "Enter the data to insert at position "
                          << position << ".\n";
                std::cin >> data;
                list.insert_spec(data, position);
                break;
            case 3:
                list.delete_beg();
                break;
            case 4:
                list.delete_end();
                break;
            case 5:
                std::cout << "Enter the position to delete data [0-n]\n";
                std::cin >> position;
                list.delete_spec(position);
                break;
            case 6:
                list.display();
                break;
            case 7:
                std::cout << "Enter the data to search\n";
                std::cin >> data;
                list.search(data);
                break;
            case 8:
                std::cout << "\nProgram ended successfully\n";
                exit(1);
                break;
        }
    }
    return 0;
}
```

# Output

```
1. Insertion at the beginning of the list

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
0

Enter the data to insert at the beginning.
10

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
0

Enter the data to insert at the beginning.
20

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
6
The data in the list is:
20 10

2. Insertion at the end of the list

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
1

Enter the data to insert at the end.
5

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
6
The data in the list is:
20 10 5

3. Insertion at a position in the list

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
2
Enter the position to insert data [0-n]
0
Enter the data to insert at position 0.
30

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
6
The data in the list is:
30 20 10 5

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
2
Enter the position to insert data [0-n]
2
Enter the data to insert at position 2.
15

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
6
The data in the list is:
30 20 15 10 5
```

4. Deletion from the beginning and end of the list

```
Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
3

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
4

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
6
The data in the list is:
20 15 10
```

5. Deletion from a position in the list

```
Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
5
Enter the position to delete data [0-n]
1

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
6
```

```
The data in the list is:
20 10
```

6. Searching for data in the list

```
Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
7
Enter the data to search
10
Data was found at position 1

Display Menu
(0) to insert at beginning,
(1) to insert at end,
(2) to insert at position,
(3) to delete from the beginning,
(4) to delete from the end,
(5) to delete from a position,
(6) to display the list,
(7) to search for data in the list
(8) to exit the program
7
Enter the data to search
999
Data is not in the list.
```