静态链表的应用

```
#include<iostream>
#include<string>
#define MaxSize 100
using namespace std;
typedef struct SListNode
{
    string data;
    int cur;
}SLinkList[MaxSize];
SLinkList s;
string str[MaxSize];
void CreatSL(SLinkList s, int n, string str[], int m)
{
    int i;
    s[0].cur = -1;
    s[1].cur = 2;
    s[0].data = "N/A";
    s[1].data = "N/A";
    int k = 0;
    for (int i = 2; i < m; i++)
    {
        if (m - n >= 2)
        {
            if (k < n)
                s[i].data = str[k++];
            else
            {
                s[i].data = "N/A";
                k++;
            }
        }
        else
        {
            s[i].data = str[k++];
        }
        s[i].cur = i + 1;
    }
    if (m != 2)
        s[m - 1].cur = -1;
    if (m - n >= 2)
    {
        if (n == 0 && m == 2)
```

```
        {
            s[0].cur = -1;
            s[1].cur = 2;
        }
        if (n != 0)
        {
            s[1].cur = s[n + 1].cur;
            s[0].cur = 2;
            s[n + 1].cur = -1;
        }
    }
    else if (m != 2)
    {
        s[0].cur = 2;
        s[1].cur = -1;
    }
    if (m == 2)
    {
        s[0].cur = -1;
        s[1].cur = -1;
    }

}
void PrintSL(SLinkList &s, int m)
{
    int  k = 0;
    for (int i = 0; i < m; i++)
    {
        cout << i << " : " << s[i].data << " : " << s[i].cur << '\t'
<< '\t';
        k++;
        if (k == 3)
        {
            cout << endl;
            k = 0;
        }
    }
    if (k != 0)
        cout << endl;
}
void PrintData(SLinkList &s)
{
    int i = 0;
    while (s[i].cur != -1)
```

```
    {
        i = s[i].cur;
        cout << s[i].data << " ";
    }
    cout << endl;
}
int Len(SLinkList &s)
{
    int i = 0;
    int len = 0;
    while (s[i].cur != -1)
    {
        i = s[i].cur;
        len++;
    }
    return len;
}
bool IsEmpty(SLinkList &s)
{
    if (s[0].cur == -1)
        return true;
    else
        return false;
}
void InsertSL(SLinkList &s, int i, string x, int &n, int m)
{
    int l, j = 0;
    int k = 1;
    if (s[k].cur == -1)
    {
        cout << "FULL" << endl;
        return;
    }
    if (IsEmpty(s) && i == 1)
    {
        s[2].data = x;
        int flag = s[s[k].cur].cur;
        s[s[k].cur].cur = -1;
        s[k].cur = flag;
        s[j].cur = 2;
        PrintData(s);
    }
    else if (i >= 1 && i <= Len(s) + 1)
    {
```

```
        s[j].cur = 2;
        for (l = 1; l < i; l++)
            j = s[j].cur;
        s[s[k].cur].data = x;
        int flag = s[s[k].cur].cur;
        s[s[k].cur].cur = s[j].cur;
        s[j].cur = s[k].cur;
        s[k].cur = flag;
        PrintData(s);
    }
    else
        cout << -1 << endl;
}
void DelSL(SLinkList &s, int j, int n, int m)
{
    int i, k = 0, l = 1;
    if (s[k].cur == -1)
        cout << "EMPTY" << endl;
    else
    {
        if (j < 1 || j>Len(s))
            cout << -1 << endl;
        else
        {
            for (i = 1; i < j; i++)
                k = s[k].cur;
            cout << s[s[k].cur].data << endl;
            int flag = s[s[k].cur].cur;
            if (s[s[1].cur].data == "N/A"&&s[1].cur != -1)
                s[s[k].cur].cur = s[1].cur;
            else
                s[s[k].cur].cur = -1;
            s[1].cur = s[k].cur;
            s[s[k].cur].data = "N/A";
            s[k].cur = flag;
        }
    }
}
void SearchSL(SLinkList &s, string z)
{
    int i = 0, count = 0, pos = 0, j = 0;
    while (j++ <= Len(s))
    {
```

```
        count++;
        if (s[i].data == z)
        {
            pos = count - 1;
            break;
        }
        i = s[i].cur;
    }
    if (pos == 0)
        cout << -1 << endl;
    else
        cout << pos << endl;
}
void AddSL(SLinkList &s, string z, int m)
{
    int k = 1;
    if (s[k].cur == -1 || m == 2)
        cout << "FULL" << endl;
    else
    {
        k = s[k].cur;
        s[1].cur = s[k].cur;
        s[k].cur = -1;
        int j = 0;
        while (s[j].cur != -1)j = s[j].cur;
        s[j].cur = k;
        s[k].data = z;
        PrintData(s);
    }
}
int main()
{
    int m, n;
    SLinkList s;
    string str[MaxSize];
    cin >> m >> n;
    for (int k = 0; k < n; k++)
        cin >> str[k];
    CreatSL(s, n, str, m);
    int i;
    string x;
    cin >> i >> x;
    int j;
    cin >> j;
```

```
    string y;
    cin >> y;
    string z;
    cin >> z;
    PrintSL(s, m);
    InsertSL(s, i, x, n, m);
    DelSL(s, j, n, m);
    SearchSL(s, y);
    AddSL(s, z, m);
    PrintSL(s, m);
    return 0;
}
```