#include <iostream>

using namespace std;

class node

{

public:

    string name;

    node \*next, \*prev;

    node()

    {

        name = "N/A";

        next = NULL;

        prev = NULL;

    }

    node(string n)

    {

        name = n;

        next = NULL;

        prev = NULL;

    }

    void productfunc(string k)

    {

        name = k;

        next = NULL;

        prev = NULL;

    }

};

class Person

{

public:

    node \*head;

    node \*curr;

    node \*n1;

    Person()

    {

        head = NULL;

    }

    Person(node \*n)

    {

        head = n;

    }

    void AddProduct(string n)

    {

        n1 = new node;

        n1->productfunc(n);

        if (head == NULL)

        {

            head = n1;

            head->next = head;

            head->prev = head;

            cout << "first person added to the cirle\n";

        }

        else

        {

            node \*last\_node = head->prev;

            last\_node->next = n1;

            n1->prev = last\_node;

            n1->next = head;

            head->prev = n1;

            cout << n1->name << "added to the circle\n";

        }

    }

    void printList()

    {

        if (!head)

        {

            cout << "List is empty" << endl;

            return;

        }

        node \*current = head;

        do

        {

            cout << current->name << " "

                 << " --> ";

            current = current->next;

        } while (current != head);

    }

    string josephus(int k)

    {

        if (!head)

        {

            cout << "List is empty, Josephus problem cannot be solved." << endl;

            return "None";

        }

        node \*current = head;

        while (current->next != current)

        {

            for (int i = 1; i < k; ++i)

            {

                current = current->next;

            }

            node \*temp = current->next;

            current->prev->next = current->next;

            current->next->prev = current->prev;

            if (head == current)

            {

                head = current->next;

            }

            delete current;

            current = temp;

            cout << "\nCurrent memebers of the Game are\n\n";

            printList();

        }

        return current->name;

    }

    ~Person()

    {

        delete n1;

    }

};

int main()

{

    Person p;

    int option;

    do

    {

        cout << "\n\n\nWhat do you want to do\n";

        cout << "1: For Adding a new Person\n";

        cout << "2: For start Elimination proccess\n";

        cout << "3: For exiting the program\n";

        cin >> option;

        if (option == 1)

        {

            string name;

            cout << "Enter the name of the person you want to add\n";

            cin >> name;

            p.AddProduct(name);

            cout << "Current persons in the circle are \n";

            p.printList();

        }

        else if (option == 2)

        {

            int num;

            cout << "Enter the number you want for elimination\n";

            cin >> num;

            string survivor;

            survivor = p.josephus(num);

            cout << "\n\nSurvivor of the current situation is " << survivor;

        }

    } while (option != 3);

    return 0;

}









