

/*Bài 29: Viết chương trình thực hiện yêu cầu sau :

Định nghĩa một cấu trúc Phân số bao gồm các trường thông tin như sau : tử số(kiểu số nguyên), mẫu số(kiểu số nguyên).

Cài đặt một cấu trúc hàng đợi bằng danh sách liên kết, với các thao tác : 1) Khởi tạo hàng đợi; 2) Kiểm tra hàng đợi rỗng;

4) Đưa một phần tử vào hàng đợi; 5) Lấy một phần tử ra khỏi hàng đợi.

Xây dựng chương trình chính để, sử dụng cấu trúc hàng đợi và các thao tác đã cài đặt trên hàng đợi để :

-Đưa vào hàng đợi n phân số bất kỳ(được nhập từ bàn phím).

- Hiển thị các phân số đã lưu trong hàng đợi và tổng của chúng ra màn hình. */

// Bài làm chưa rút gọn Phân Số

```
#include<iostream>
```

```
using namespace std;
```

```
struct PhanSo { // Khởi tạo tử = 0, mẫu = 1 để tính tổng các PS
```

```
    int TuSo = 0;
```

```
    int MauSo = 1;
```

```
};
```

```
struct NODE { // Cấu trúc NODE
```

```
    PhanSo DaTa;
```

```
    NODE* pNext;
```

```
};
```

```
struct QUEUE { // Cấu trúc QUEUE
```

```
    // Front là giá trị đầu hàng đợi, Rear là giá trị cuối
```

```
    NODE* Front;
```

```
    NODE* Rear;
```

```
};
```

```
void KhoiTao(QUEUE& Q) { // Khởi Tạo QUEUE rỗng
```

```
    Q.Front = Q.Rear = NULL;
```

```
}
```

```
bool KiemTraRong(QUEUE Q) { // Kiểm tra rỗng
```

```
    if (Q.Rear == NULL) {
```

```
        return true;
```

```
    }
```

```
    else return false;
```

```
}
```

```
NODE* TaoMoiNODE(PhanSo PS) { // Tạo Mới Nút Chứa Phân Số
```

```
    NODE* K = new NODE;
```

```
    if (K == NULL) {
```

```
        cout << "KHONG CAP PHAT DUOC BO NHO !" << endl;
```

```
        return NULL;
```

```

    }
    K->DaTa = PS;
    K->pNext = NULL;
    return K;
}

istream& operator >> (istream& input, PhanSo& PS) { // Nạp chồng Nhập
    cout << "Nhap Tu So : " << endl;
    input >> PS.TuSo;
    do {
        cout << "Nhap Mau So : " << endl;
        input >> PS.MauSo;
        if (PS.MauSo <= 0) {
            cout << "BAN NHAP SAI MAU SO ! MOI BAN NHAP LAI
MAU SO !" << endl;
        }
    } while (PS.MauSo <= 0);
    return input;
}

ostream& operator << (ostream& output, PhanSo PS) { // Nạp chồng xuất
    output << PS.TuSo << "/" << PS.MauSo << endl;
    return output;
}

PhanSo operator + (PhanSo &PS1, PhanSo &PS2) { // Nạp chồng toán tử + 2 PS
    PhanSo PS3;
    PS3.TuSo = PS1.TuSo * PS2.MauSo + PS1.MauSo * PS2.TuSo;
    PS3.MauSo = PS1.MauSo * PS2.MauSo;
    return PS3;
}

void ADD(QUEUE &Q, PhanSo PS){ // Hàm thêm vào hàng đợi
    NODE* K = TaoMoiNODE(PS); // Khởi tạo nút K chứa Dữ liệu là Phân Số
    if (KiemTraRong(Q) == true) {
        Q.Front = Q.Rear = K;
    }
    else { // Thêm vào cuối hàng đợi
        Q.Rear->pNext = K;
        Q.Rear = K;
    }
}

```

```

PhanSo REMOVE(QUEUE& Q) { // Hàm lấy ra Phân Số ở Hàng đợi
    PhanSo PS;
    if (KiemTraRong(Q) == true) {
        cout << "HANG DOI DA RONG !" << endl;
    }
    else {
        NODE* K = Q.Front; // Ghi nhận phần tử đầu danh sách và giá trị
của nút
        PS = K->DaTa;
        Q.Front = Q.Front->pNext; // điều chỉnh liên kết
        if (Q.Front == NULL) {
            Q.Rear = NULL;
        }
        K->pNext = NULL; // cô lập nút k
        delete K; // xóa nút k
        return PS; // trả về Phân Số
    }
}

int main() {
    QUEUE Q;
    PhanSo PS, S;
    int n;
    KhoiTao(Q);
    cout << "NHAP SO LUONG PHAN SO : " << endl;
    cin >> n;
    for (int i = 0; i < n; i++) {
        cout << "Nhap PS " << i + 1 << ": " << endl;
        cin >> PS;
        ADD(Q, PS);
    }
    while (KiemTraRong(Q) == false) {
        PS = REMOVE(Q);
        cout << PS << endl;
        S = S + PS; // Lấy ra lần lượt từng PS và Tính Tổng chúng lun
    }
    cout << "Tong Cac Phan So: " << S << endl;
    system("pause");
    return 0;
}

```

```
D:\CTDLGT\Bai29\Debug\Bai29.exe
NHAP SO LUONG PHAN SO :
3
Nhap PS 1:
Nhap Tu So :
1
Nhap Mau So :
2
Nhap PS 2:
Nhap Tu So :
1
Nhap Mau So :
3
Nhap PS 3:
Nhap Tu So :
1
Nhap Mau So :
4
1/2
1/3
1/4
Tong Cac Phan So: 26/24
Press any key to continue . . .
```