```
/*********************************************************************************
  *Copyright(C)  北京邮电大学智能科学与技术 2016211319 班  刘畅
  *FileName:     LRU.cpp, FIFO.cpp, Optimal.cpp
  *Author:       刘畅
  *Version:       1.0
  *Date:          2018.12.18
  *Description:   模拟调页算法 LRU, FIFO, Optimal
  *Compile:       g++ -std=c++17 LRU.cpp -o LRU
  *Function List:
     1. LRU             2. FIFO         3. Optimal
  *Key Parameters:
     1. sequence     页的引用串
     2. frame        帧数
     3. buffer       页表, 长度为  frame
     4. cnt          计数, 页表对应位置未被使用的时间
     5. buffer       页表, 长度为  frame
     6. fault        当前页错误个数
  *History:
     1.Date:         2018.12.18
       Author:       刘畅
       Modification: 完成了基本功能
       Problem:
*********************************************************************************/
#include <iostream>
#include <algorithm>
#define all(x) x.begin(), x.end()

void LRU(const std::vector<int>& sequence, const int frame) {
    int fault = 0;
    std::vector<int> buffer(frame, -1);
    std::vector<int> cnt(frame, 0);
    for(const auto it : sequence) {
        for(auto &it : cnt) it += 1;
        auto pos = std::distance(buffer.begin(), std::find(all(buffer), it));
        if(pos == frame) {
            ++fault;
            // 现在开始找计数最大的，替换出去
            pos = std::distance(cnt.begin(), std::max_element(all(cnt)));
            buffer[pos] = it;
        }
        cnt[pos] = 0;
    }
    std::cout << "LRU      : " << fault << "\n";
}
```

```
void FIFO(const std::vector<int>& sequence, const int frame) {
    int fault = 0;
    std::vector<int> buffer(frame, -1);
    for(auto cur = sequence.begin();cur not_eq sequence.end(); ++cur) {
        if(std::find(all(buffer), *cur) == buffer.end()) {
            buffer[fault % frame] = *cur;
            ++fault;
        }
    }
    std::cout << "FIFO     : " << fault << "\n";
}

void Optimal(const std::vector<int>& sequence, const int frame) {
    int fault = 0;
    std::vector<int> buffer(frame, -1);
    for(auto cur = sequence.begin();cur not_eq sequence.end(); ++cur) {
        // 如果当前缺页
        if(std::find(all(buffer), *cur) == buffer.end()) {
            auto late = -1, M = -1;
            for(int i = 0;i < frame; ++i) {
        auto dis = std::distance(cur, std::find(cur, sequence.end(), buffer[i]));
                if(dis > M) M = dis, late = i;
            }
            buffer[late] = *cur;
            ++fault;
        }
    }
    std::cout << "Optimal  : " << fault << "\n";
}

int main() {
    const int frame = 3;
    std::vector<int> sequence{7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7,1,0, 5,4,6, 2, 3, 0,1};
    LRU(sequence, frame);
    FIFO(sequence, frame);
    Optimal(sequence, frame);
    return 0;
}

/* 运行结果
LRU      : 18
FIFO     : 17
Optimal  : 13
*/
```