```cpp
//OS Lab Assignment: Page Replacement Algorithm.
//By Sanskar Sharma
//PRN: 0120180381
//Roll Number: 090

#include <bits/stdc++.h>
#include <set>
#include <unordered_set>
using namespace std;

void print_queue(queue<int> q)
{
    while (!q.empty())
    {
        cout << q.front()<<" < ";
        q.pop();
    }
    cout << endl;
}

bool checkValue(queue<int> q, int temp)
{
    bool t = 0;
    while (!q.empty())
    {
        if(q.front() == temp){
            t = 1;
            break;
        }
        else{
            q.pop();
        }
    }
    return t;
}

//FIFO Page Replacement Algorithm
void FIFO(){
    int index = -1;
    int n, frame, temp,pagef=0,hits=0;;
    cout<<"\t\t\tEnter the number of reference & frame slots (separated by spaces): ";
    cin>>n>>frame;
    queue <int> q;
    cout<<"\t\t\tEnter the page reference in sequence (separated by spaces): \n\t\t\t  ";
    for(int i=0;i<n;i++){
        cin>>temp;
        if(checkValue(q, temp)){
            hits++;
            continue;
        }
        else{
            pagef++;
            if(index>=frame-1){
                q.pop();
                q.push(temp);
            }
            else{
                q.push(temp);
                index++;
            }
        }
        // print_queue(q);
        // cout<<"Index: "<<index<<endl;
    }
    cout<<"\n\tPage Faults : "<<pagef<<
          "\n\tHits        : "<<hits<<"\n";


}

bool search(int key, vector<int>& fr)
{
    for (int i = 0; i < fr.size(); i++)
        if (fr[i] == key)
            return true;
    return false;
}

int predict(int pg[], vector<int>& fr, int pn, int index)
{
    int res = -1, farthest = index;
    for (int i = 0; i < fr.size(); i++) {
        int j;
        for (j = index; j < pn; j++) {
            if (fr[i] == pg[j]) {
                if (j > farthest) {
                    farthest = j;
                    res = i;
                }
                break;
            }
        }
        if (j == pn)
            return i;
    }
    retuif(q.front() == temp){es;
}

//Optimal Page Replacement Algorithm.
void Optimal()
{
    vector<int> F;
    int hit = 0;
    int n,frame;
    cout<<"\t\t\tEnter the number of reference & frame slots (separated by spaces): ";
    cin>>n>>frame;
    int page[n];
    cout<<"\t\t\tEnter the page reference in sequence (separated by spaces): \n\t\t\t  ";
    for(int i=0;i<n;i++){
        cin>>page[i];
    }

    for (int i=0;i<n;i++) {

        if (search(page[i], F)) {
            hit++;
            continue;
        }
        if (F.size() < frame)
            F.push_back(page[i]);
        else {
            int j = predict(page, F, n, i + 1);
            F[j] = page[i];
        }
    }
    cout<<"\n\tPage Faults : "<<n-hit<<
          "\n\tHits        : "<<hit<<"\n";
}

//Least Recently Used Algorithm.
void Least()
{
    int n, frame,pagef=0;
    cout<<"\t\t\tEnter the number of reference & frame slots (separated by spaces): ";
    cin>>n>>frame;
    int pages[n];
    cout<<"\t\t\tEnter the page reference in sequence (separated by spaces): \n\t\t\t  ";
    for(int i=0;i<n;i++){
        cin>>pages[i];
    }
    unordered_set <int> s;
    unordered_map<int, int> indexes;
    for (int i=0; i<n; i++)
    {
        if (s.size() < frame)
        {
            if (s.find(pages[i])==s.end())
            {
                s.insert(pages[i]);
                pagef++;
            }
            indexes[pages[i]] = i;
        }

        else
        {
            if (s.find(pages[i]) == s.end())
            {
                int lru = INT_MAX, val;
                for (auto it=s.begin(); it!=s.end(); it++)
                {
                    if (indexes[*it] < lru)
                    {
                        lru = indexes[*it];
                        val = *it;
                    }
                }
                s.erase(val);
                s.insert(pages[i]);
                pagef++;
            }
            indexes[pages[i]] = i;
        }
    }
    cout<<"\n\tPage Faults : "<<pagef<<
          "\n\tHits        : "<<n-pagef<<"\n";
}

void PageReplacement()
{
    cout<<"\n\t\tOS Lab Assignment: Page Replacement Algorithm";
    cout<<"\n\t\t\tBy Sanskar Sharma 090";
    cout<<"\n\t\t\t\t   PRN: 0120180381";
    cout<<"\n\tChoose the Page Replacement algorithm.\n";
    cout<<"\t\t1. FIFO\n"<<
          "\t\t2. Optimal Page Replacement.\n"<<
          "\t\t3. Least Recently Used\n"<<
          "\t\t4. EXIT\n"<<
          "\t\tEnter your choice: ";
    int choice;
    cin>>choice;
    switch(choice){
        case 1:
            FIFO();
            PageReplacement();
            break;
        case 2:
            Optimal();
            PageReplacement();
            break;
        case 3:
            Least();
            PageReplacement();
            break;
        case 4:
            exit(0);
            break;
        default:
            exit(-1);//entered wrong choice
    }
}

int main(){
    system("clear");

    //Page Replacement Algorithms.
    PageReplacement();

    return 0;
}


/*
Output of Page Replacement Algorithm:

        OS Lab Assignment: Page Replacement Algorithm
            By Sanskar Sharma 090
               PRN: 0120180381
    Choose the Page Replacement algorithm.
        1. FIFO
        2. Optimal Page Replacement.
        3. Least Recently Used
        4. EXIT
        Enter your choice: 1
            Enter the number of reference & frame slots (separated by spaces): 7 3
            Enter the page reference in sequence (separated by spaces):
              1 3 0 3 5 6 3

    Page Faults : 6
    Hits        : 1
        OS Lab Assignment: Page Replacement Algorithm
            By Sanskar Sharma 090
               PRN: 0120180381
    Choose the Page Replacement algorithm.
        1. FIFO
        2. Optimal Page Replacement.
        3. Least Recently Used
        4. EXIT
        Enter your choice: 2
            Enter the number of reference & frame slots (separated by spaces): 14 4
            Enter the page reference in sequence (separated by spaces):
              7 0 1 2 0 3 0 4 2 3 0 3 2 3

    Page Faults : 6
    Hits        : 8

        OS Lab Assignment: Page Replacement Algorithm
            By Sanskar Sharma 090
               PRN: 0120180381
    Choose the Page Replacement algorithm.
        1. FIFO
        2. Optimal Page Replacement.
        3. Least Recently Used
        4. EXIT
        Enter your choice: 3
            Enter the number of reference & frame slots (separated by spaces): 12 4
            Enter the page reference in sequence (separated by spaces):
              1 2 3 4 5 1 3 1 6 3 2 3

    Page Faults : 8
    Hits        : 4

        OS Lab Assignment: Page Replacement Algorithm
            By Sanskar Sharma 090
               PRN: 0120180381
    Choose the Page Replacement algorithm.
        1. FIFO
        2. Optimal Page Replacement.
        3. Least Recently Used
        4. EXIT
        Enter your choice: 4

...Program finished with exit code 0
Press ENTER to exit console.

*/
```