**First In First Out(FIFO)**:

#include <iostream>

#include <list>

#include <unordered\_set>

using namespace std;

class FIFOPageReplacement {

private:

list<int> pageQueue;

unordered\_set<int> pageSet;

int capacity;

int pageHits;

int pageMisses;

public:

FIFOPageReplacement(int capacity) {

this->capacity = capacity;

pageHits = 0;

pageMisses = 0;

}

void referencePage(int page) {

if (pageSet.find(page) != pageSet.end()) {

pageHits++;

return;

}

if (pageQueue.size() == capacity) {

int frontPage = pageQueue.front();

pageQueue.pop\_front();

pageSet.erase(frontPage);

}

pageMisses++;

pageQueue.push\_back(page);

pageSet.insert(page);

}

int getPageHits() {

return pageHits;

}

float getPageMisses() {

return pageMisses;

}

};

int main() {

int capacity;

cout << "Enter the number of frames: ";

cin >> capacity;

FIFOPageReplacement fifo(capacity);

int n;

cout << "Enter the number of page references: ";

cin >> n;

cout << "Enter the page references: ";

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

int page;

cin >> page;

fifo.referencePage(page);

}

cout << "Page Hits: " << fifo.getPageHits() << endl;

cout << "Page faults: " << fifo.getPageMisses() << endl;

// cout<<"Average Page faults<<"(fifo.getPageMisses()/n)\*100;

return 0;

}

**OUTPUT**:

Enter the number of frames: 3

Enter the number of page references: 15

Enter the page references: 7 0 1 2 0 3 0 4 2 3 0 3 1 2 0

Page Hits: 3

Page faults: 12