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

#include <iostream>

using namespace std;

int main() {

int frameSize, streamLength;

cout << "Enter the frame size: ";

cin >> frameSize;

cout << "Enter the page reference stream length: ";

cin >> streamLength;

int frames[frameSize];

int pageStream[streamLength];

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

frames[i] = -1;

}

cout << "Enter the page reference stream:" << endl;

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

cin >> pageStream[i];

}

int pageHits = 0;

int pageFaults = 0;

int replacePosition = 0;

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

int found = 0;

for (int j = 0; j < frameSize; j++) {

if (frames[j] == pageStream[i]) {

found = 1;

pageHits++;

break;

}

}

if (found == 0) {

frames[replacePosition] = pageStream[i];

replacePosition = (replacePosition + 1) % frameSize;

pageFaults++;

}

}

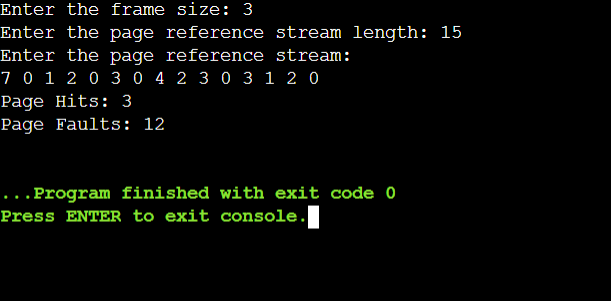
cout << "Page Hits: " << pageHits << endl;

cout << "Page Faults: " << pageFaults << endl;

return 0;

}

**Output:**

****

**Least Recently Used (LRU):**

#include <iostream>

using namespace std;

int main() {

int frameSize, pageStreamLength, hits = 0, faults = 0;

cout << "Enter frame size: ";

cin >> frameSize;

cout << "Enter page reference stream length: ";

cin >> pageStreamLength;

int pages[pageStreamLength], frames[frameSize], lastUsed[frameSize];

cout << "Enter page reference stream: ";

for (int i = 0; i < pageStreamLength; i++) cin >> pages[i];

for (int i = 0; i < frameSize; i++) frames[i] = -1, lastUsed[i] = -1;

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

int j;

for (j = 0; j < frameSize; j++) {

if (frames[j] == pages[i]) {

hits++;

lastUsed[j] = i;

break;

}

}

if (j == frameSize) {

int lru = 0;

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

if (lastUsed[k] < lastUsed[lru]) lru = k;

frames[lru] = pages[i];

lastUsed[lru] = i;

faults++;

}

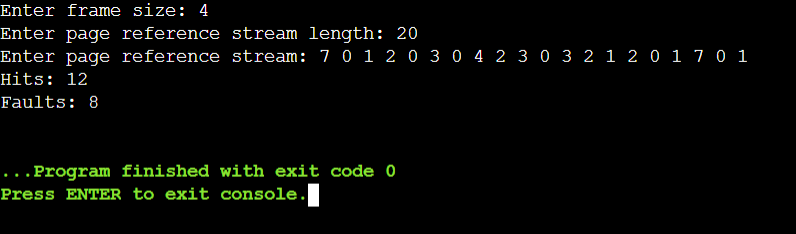
}

cout << "Hits: " << hits << "\nFaults: " << faults << endl;

return 0;

}

**Output:**

****

**Optimal:**

#include <iostream>

using namespace std;

int main() {

int frameSize, pagesLength, pages[100], frames[10], pageHits = 0, pageFaults = 0;

cout << "Enter frame size: ";

cin >> frameSize;

cout << "Enter length of page reference stream: ";

cin >> pagesLength;

for(int i = 0; i < frameSize; i++) frames[i] = -1;

cout << "Enter page reference stream:\n";

for(int i = 0; i < pagesLength; i++) cin >> pages[i];

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

int found = 0, replace = 0, farthest = i;

for(int j = 0; j < frameSize; j++) {

if(frames[j] == pages[i]) {

pageHits++;

found = 1;

break;

}

int k;

for(k = i; k < pagesLength; k++) if(frames[j] == pages[k]) break;

if(k > farthest) { farthest = k; replace = j; }

}

if(!found) {

frames[replace] = pages[i];

pageFaults++;

}

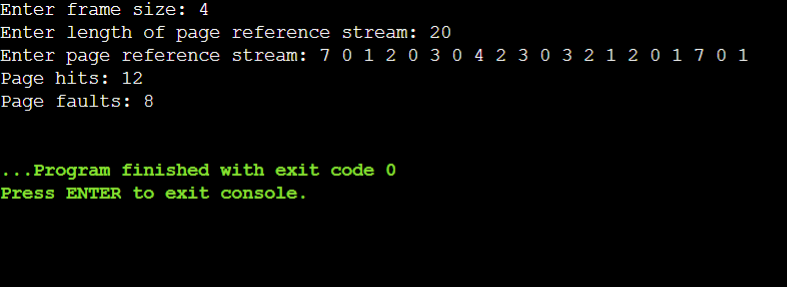
}

cout << "Page hits: " << pageHits << "\nPage faults: " << pageFaults << endl;

return 0;

}

**Output:**

****