Assignment 7

Name: Sujal Santosh Porte

PRN: 122B1B227

```
#include <iostream>
#include <vector>
#include <unordered_map>
#include <unordered_set>
#include <queue>
#include <algorithm>
using namespace std;
void inputReferenceString(vector<int>& refString, int& frames) {
int n;
cout << "Enter number of frames: ";
cin >> frames;
cout << "Enter length of reference string: ";
cin >> n;
refString.resize(n);
cout << "Enter the reference string:\n";
for (int i = 0; i < n; ++i)
cin >> refString[i];
}
void displayFrames(const vector<int>& memory) {
cout << "\nFinal Frame Content:\n";
for (int i = 0; i < memory.size(); ++i) {
cout << "Frame " << i + 1 << ": " << memory[i] << "\n";
}
}
void FCFS(const vector<int>& refString, int frames) {
unordered_set<int> memorySet;
queue<int> order;
vector<int> memory;
for (int page : refString) {
if (memorySet.find(page) == memorySet.end()) {
if (memorySet.size() == frames) {
int old = order.front();
order.pop();
memorySet.erase(old);
auto it = find(memory.begin(), memory.end(), old);
if (it != memory.end())
*it = page;
} else {
memory.push_back(page);
```

```
}
memorySet.insert(page);
order.push(page);
}
void LRU(const vector<int>& refString, int frames) {
unordered_map<int, int> lastUsed;
vector<int> memory;
for (int i = 0; i < refString.size(); ++i) {
int page = refString[i];
bool found = false;
for (int m : memory) {
if (m == page) {
found = true;
break;
}
}
if (!found) {
if (memory.size() < frames) {
memory.push_back(page);
} else {
int lruIndex = 0;
for (int j = 1; j < memory.size(); ++j) {
if (lastUsed[memory[j]] < lastUsed[memory[lruIndex]])
lruIndex = j;
}
memory[lruIndex] = page;
}
}
lastUsed[page] = i;
}
cout << "\n--- LRU Page Replacement ---\n";
displayFrames(memory);
}
void Optimal(const vector<int>& refString, int frames) {
vector<int> memory;
for (int i = 0; i < refString.size(); ++i) {
int page = refString[i];
if (find(memory.begin(), memory.end(), page) == memory.end()) {
if (memory.size() < frames) {
```

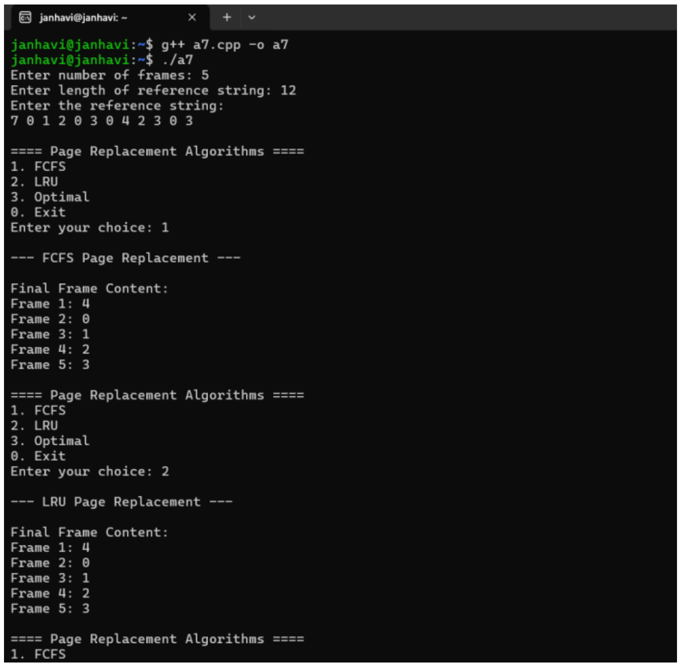
```
}
cout << "\n--- FCFS Page Replacement ---\n";
displayFrames(memory);
}
```

```
memory.push_back(page);
} else {
int indexToReplace = -1, farthest = i + 1;
for (int j = 0; j < memory.size(); ++j) {
int k;
for (k = i + 1; k < refString.size(); ++k) {
if (refString[k] == memory[j]) break;
}
if (k > farthest) {
farthest = k;
indexToReplace = j;
}
if (k == refString.size()) {
indexToReplace = j;
break;
}
}
if (indexToReplace == -1)
indexToReplace = 0;
memory[indexToReplace] = page;
}
}
}
cout << "\n--- Optimal Page Replacement ---\n";
displayFrames(memory);
}
```

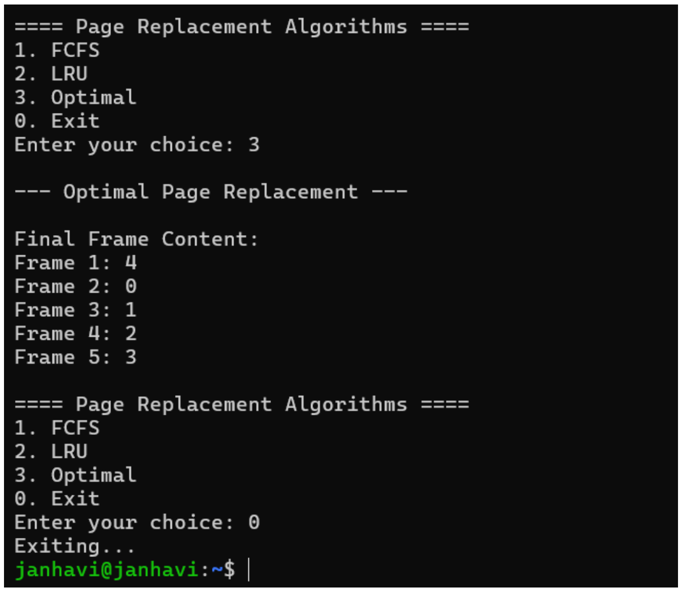
```
int main() {
vector<int> refString;
int frames, choice;
inputReferenceString(refString, frames);
do {
cout << "\n==== Page Replacement Algorithms ====\n";
cout << "1. FCFS\n";
cout << "2. LRU\n";
cout << "3. Optimal\n";
cout << "0. Exit\n";
cout << "Enter your choice: ";
cin >> choice;
switch (choice) {
case 1:
FCFS(refString, frames);
break;
case 2:
```

```
LRU(refString, frames);
break;
case 3:
Optimal(refString, frames);
break;
case 0:
cout << "Exiting...\n";
break;
default:
cout << "Invalid choice!\n";
}
} while (choice != 0);
return 0;
}
```

OUTPUT:



```
janhavi@janhavi:~$ g++ a7.cpp -o a7
janhavi@janhavi:~$ ./a7
Enter number of frames: 5
Enter length of reference string: 12
Enter the reference string:
7 0 1 2 0 3 0 4 2 3 0 3

==== Page Replacement Algorithms ====
1. FCFS
2. LRU
3. Optimal
0. Exit
Enter your choice: 1

--- FCFS Page Replacement ---

Final Frame Content:
Frame 1: 4
Frame 2: 0
Frame 3: 1
Frame 4: 2
Frame 5: 3

==== Page Replacement Algorithms ====
1. FCFS
2. LRU
3. Optimal
0. Exit
Enter your choice: 2

--- LRU Page Replacement ---

Final Frame Content:
Frame 1: 4
Frame 2: 0
Frame 3: 1
Frame 4: 2
Frame 5: 3

==== Page Replacement Algorithms ====
1. FCFS
```



```
==== Page Replacement Algorithms ====
1. FCFS
2. LRU
3. Optimal
0. Exit
Enter your choice: 3

--- Optimal Page Replacement ---

Final Frame Content:
Frame 1: 4
Frame 2: 0
Frame 3: 1
Frame 4: 2
Frame 5: 3

==== Page Replacement Algorithms ====
1. FCFS
2. LRU
3. Optimal
0. Exit
Enter your choice: 0
Exiting...
janhavi@janhavi:~$ 
```