EXPERIMENT-5.2

Aim:

Wite a program for page replacement policy using Least Recently Use (LRU) algorithm.

Theory:

Code:

```
#include <bits/stdc++.h>

using namespace std;

void printList(list<int> l) {
    for (int page : l) {
        cout << page << " ";
    }
}

int main() {
    int frameSize, numReferences;

    cout << "Enter the number of frames: ";
    cin >> frameSize;

    cout << "Enter the number of page references: ";
    cin >> numReferences;

    cout << "Enter the page reference string: ";
    vector<int> pageReferences(numReferences);
    for (int i = 0; i < numReferences; i++) {
        cin >> pageReferences[i];
    }

    list<int> lruList;
    unordered_set<int> pageSet;
    int pageFaults = 0;

    cout << "\nPage Insertion Order:" << endl;

    for (int i = 0; i < numReferences; i++) {
        int currentPage = pageReferences[i];

        if (pageSet.find(currentPage) == pageSet.end()) {
            if (lruList.size() == frameSize) {
                int leastRecentlyUsedPage = lruList.back();
                lruList.pop_back();
                pageSet.erase(leastRecentlyUsedPage);
            }

            lruList.push_front(currentPage);
            pageSet.insert(currentPage);
            pageFaults++;

            cout << "Page " << currentPage << " inserted. ";
            cout << "Current List: ";
```

```
            printList(lruList);
            cout << endl;
        } else {
            lruList.remove(currentPage);
            lruList.push_front(currentPage);

            cout << "Page " << currentPage << " already in frame. ";
            cout << "Current List: ";
            printList(lruList);
            cout << endl;
        }
    }

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

    return 0;
}
```

Output:

```
PS C:\Users\ankus\OneDrive\Desktop\test> ./LRU.exe
Enter the number of frames: 3
Enter the number of page references: 7
Enter the page reference string: 1 3 0 3 5 6 3

Page Insertion Order:
Page 1 inserted. Current List: 1
Page 3 inserted. Current List: 3 1
Page 0 inserted. Current List: 0 3 1
Page 3 already in frame. Current List: 3 0 1
Page 5 inserted. Current List: 5 3 0
Page 6 inserted. Current List: 6 5 3
Page 3 already in frame. Current List: 3 6 5

Total Page Faults: 5
```

EXPERIMENT-5.3

Aim:

Wite a program for page replacement policy using Optimal algorithm.

Theory:

Code:

```
#include <bits/stdc++.h>

using namespace std;

void printFrames(vector<int> frames) {
    for (int page : frames) {
        cout << page << " ";
    }
}

int main() {
    int frameSize, numReferences;

    cout << "Enter the number of frames: ";
    cin >> frameSize;

    cout << "Enter the number of page references: ";
    cin >> numReferences;

    cout << "Enter the page reference string: ";
    vector<int> pageReferences(numReferences);
    for (int i = 0; i < numReferences; i++) {
        cin >> pageReferences[i];
    }

    vector<int> frames(frameSize, -1);
    int pageFaults = 0;

    cout << "\nPage Insertion Order:" << endl;

    for (int i = 0; i < numReferences; i++) {
        int currentPage = pageReferences[i];

        if (find(frames.begin(), frames.end(), currentPage) == frames.end()) {
            int indexToReplace = -1;
            int farthestReference = -1;

            for (int j = 0; j < frameSize; j++) {
                int nextPage = pageReferences.size();
                for (int k = i + 1; k < numReferences; k++) {
                    if (frames[j] == pageReferences[k]) {
                        nextPage = k;
                        break;
                    }
                }
                if (nextPage > farthestReference) {
                    farthestReference = nextPage;
```

```
                    indexToReplace = j;
                }
            }
            frames[indexToReplace] = currentPage;
            pageFaults++;

            cout << "Page " << currentPage << " inserted. ";
            cout << "Current Frames: ";
            printFrames(frames);
            cout << endl;
        } else {
            cout << "Page " << currentPage << " already in frame. ";
            cout << "Current Frames: ";
            printFrames(frames);
            cout << endl;
        }
    }
    cout << "\nTotal Page Faults: " << pageFaults << endl;

    return 0;
}
```

Output:

```
PS C:\Users\ankus\OneDrive\Desktop\test> ./optimal.exe
Enter the number of frames: 3
Enter the number of page references: 20
Enter the page reference string: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Page Insertion Order:
Page 7 inserted. Current Frames: 7 -1 -1
Page 0 inserted. Current Frames: 7 0 -1
Page 1 inserted. Current Frames: 7 0 1
Page 2 inserted. Current Frames: 2 0 1
Page 0 already in frame. Current Frames: 2 0 1
Page 3 inserted. Current Frames: 2 0 3
Page 0 already in frame. Current Frames: 2 0 3
Page 4 inserted. Current Frames: 2 4 3
Page 2 already in frame. Current Frames: 2 4 3
Page 3 already in frame. Current Frames: 2 4 3
Page 0 inserted. Current Frames: 2 0 3
Page 3 already in frame. Current Frames: 2 0 3
Page 2 already in frame. Current Frames: 2 0 3
Page 1 inserted. Current Frames: 2 0 1
Page 2 already in frame. Current Frames: 2 0 1
Page 0 already in frame. Current Frames: 2 0 1
Page 1 already in frame. Current Frames: 2 0 1
Page 7 inserted. Current Frames: 7 0 1
Page 0 already in frame. Current Frames: 7 0 1
Page 1 already in frame. Current Frames: 7 0 1

Total Page Faults: 9
```