# Bansilal Ramnath Agarwal Charitable Trust’s

Vishwakarma Institute of Technology, Pune-37

*(Anautonomous Institute of Savitribai Phule Pune University)*



**Department of Computer Engineering**

|  |  |
| --- | --- |
| **Division** | **CS** |
| **Batch** | **B1** |
| **Roll no.** | **90** |
| **Name** | **Aditya Shrinivas Kurapati** |
| **PRN No** | **12320184** |

* **Page Replacement Policies**

#include <stdio.h>

#include <stdbool.h>

#define MAX\_FRAMES 3

// Structure to represent a page in memory

typedef struct {

int page\_number;

int last\_used; // Timestamp to track when the page was last used

bool valid;

} Page;

// Function to find the least recently used page

int findLRU(Page frames[], int num\_frames) {

int lru\_index = 0;

for (int i = 1; i < num\_frames; ++i) {

if (frames[i].last\_used < frames[lru\_index].last\_used) {

lru\_index = i;

}

}

return lru\_index;

}

// Function to simulate page replacement using LRU policy

void LRU(int pages[], int num\_pages) {

Page frames[MAX\_FRAMES];

int page\_faults = 0;

for (int i = 0; i < num\_pages; ++i) {

bool page\_found = false;

// Check if page is already in memory

for (int j = 0; j < MAX\_FRAMES; ++j) {

if (frames[j].valid && frames[j].page\_number == pages[i]) {

frames[j].last\_used = i; // Update timestamp

page\_found = true;

break;

}

}

// If page not found in memory, perform page replacement

if (!page\_found) {

int lru\_index = findLRU(frames, MAX\_FRAMES);

frames[lru\_index].page\_number = pages[i];

frames[lru\_index].last\_used = i;

frames[lru\_index].valid = true;

page\_faults++;

}

}

printf("Number of page faults using LRU: %d\n", page\_faults);

}

int main() {

int pages[] = {0, 1, 2, 3, 2, 4, 5, 3, 4}; // Sample page reference string

int num\_pages = sizeof(pages) / sizeof(pages[0]);

LRU(pages, num\_pages);

return 0;

}

