B. Siva Shirish-192324016

32. Construct a C program to simulate the Least Recently Used paging technique of memory management.

AIM

To construct a C program that simulates the Least Recently Used (LRU) paging technique of memory management, which replaces the page that has not been used for the longest time when a new page needs to be loaded, and all frames are full.

ALGORITHM

1. **Start**
2. Input the number of pages, the sequence of page references, and the number of frames.
3. Initialize the frames with -1 (empty), and set the page fault counter to 0.
4. For each page reference:
    - Check if the page is already present in one of the frames.
        - If it is found, move to the next page (no page fault).
        - If it is not found, increment the page fault counter.
        - If there is space in the frames, place the page in an empty frame.
        - If all frames are full, find the least recently used page (the one that hasn't been used for the longest time) and replace it with the new page.
5. Display the status of the frames after each page reference and the total number of page faults at the end.
6. **Stop**

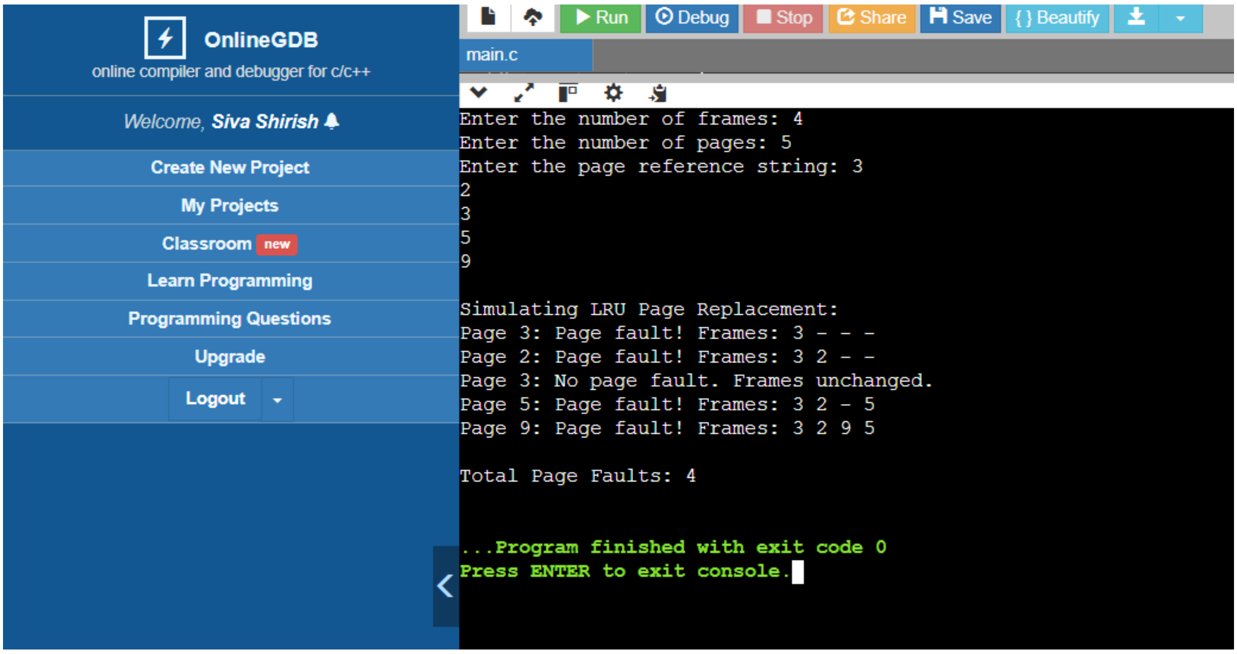
PROCEDURE

1. Include necessary libraries (stdio.h for input and output).
2. Define a function lruPaging() to simulate the LRU paging technique:
    - Initialize an array to represent the frames and set all elements to -1.
    - Iterate over each page in the page reference sequence and check if it is in the frames.
    - If the page is found, update the frame with the new reference and continue.
    - If the page is not found, determine which page has been used least recently, and replace it.
3. Input the number of pages, the reference sequence, and the number of frames from the user.
4. Call the lruPaging() function and display the frame status after each page reference.
5. Print the total number of page faults at the end.

CODE:

```
#include <stdio.h>

#define MAX_FRAMES 10
#define MAX_PAGES 50

int findLRU(int recent[], int numFrames) {
    int minIndex = 0;
    for (int i = 1; i < numFrames; i++) {
        if (recent[i] < recent[minIndex]) {
            minIndex = i;
        }
    }
    return minIndex;
}

int main() {
    int frames[MAX_FRAMES], pages[MAX_PAGES], recent[MAX_FRAMES];
    int numFrames, numPages, pageFaults = 0, time = 0;

    printf("Enter the number of frames: ");
    scanf("%d", &numFrames);


    printf("Enter the number of pages: ");
    scanf("%d", &numPages);

    printf("Enter the page reference string: ");
    for (int i = 0; i < numPages; i++) {
        scanf("%d", &pages[i]);
    }


    for (int i = 0; i < numFrames; i++) {
        frames[i] = -1;
        recent[i] = 0;
    }

    printf("\nSimulating LRU Page Replacement:\n");

    for (int i = 0; i < numPages; i++) {
        int page = pages[i];
        int found = 0;

        for (int j = 0; j < numFrames; j++) {
            if (frames[j] == page) {
                found = 1;
                recent[j] = time++;
                break;
            }
        }

        if (!found) {
            pageFaults++;

            int pos;
            if (i < numFrames) {
                pos = i;
            } else {
                pos = findLRU(recent, numFrames);
            }

            frames[pos] = page;
            recent[pos] = time++;


            printf("Page %d: Page fault! Frames: ", page);
            for (int j = 0; j < numFrames; j++) {
                if (frames[j] == -1) printf("- ");
                else printf("%d ", frames[j]);
            }
            printf("\n");
        } else {
            printf("Page %d: No page fault. Frames unchanged.\n", page);
        }
    }


    printf("\nTotal Page Faults: %d\n", pageFaults);
    return 0;
}
```

OUTPUT:



```
Enter the number of frames: 4
Enter the number of pages: 5
Enter the page reference string: 3
2
3
5
9

Simulating LRU Page Replacement:
Page 3: Page fault! Frames: 3 - - -
Page 2: Page fault! Frames: 3 2 - -
Page 3: No page fault. Frames unchanged.
Page 5: Page fault! Frames: 3 2 - 5
Page 9: Page fault! Frames: 3 2 9 5

Total Page Faults: 4


...Program finished with exit code 0
Press ENTER to exit console.
```