B. Siva Shirish-192324016

31. Construct a C program to simulate the First in First Out paging technique of memory management.

AIM

To construct a C program to simulate the First-In-First-Out (FIFO) paging technique of memory management, which replaces the oldest page in memory when a new page needs to be loaded and all frames are full.

ALGORITHM

1. Start
2. Input the total number of pages, the sequence of page references, and the number of available frames.
3. Initialize the frames as empty (-1) and set the page fault counter to 0.
4. For each page in the reference sequence:
    - Check if the page is already present in any of the frames.
    - If the page is found in the frames, move to the next page (no page fault).
    - If the page is not found:
        - Replace the oldest page in the frames using the FIFO approach.
        - Increment the page fault counter.
    - Update the frame contents and display the current frame status.
5. Display the total number of page faults after processing all pages.
6. Stop

PROCEDUR

E

1. Include necessary libraries.
2. Define variables for frame size, pages, page faults, and an array to represent frames.
3. Take input for the number of pages, the sequence of page references, and the number of frames.
4. Use a loop to process each page reference in the sequence:

- Check if the page is already in a frame.
- If not, replace the oldest page in the frame using the FIFO technique.
- Increment the page fault counter.

5. Display the frame status after each page reference.
6. Print the total number of page faults.

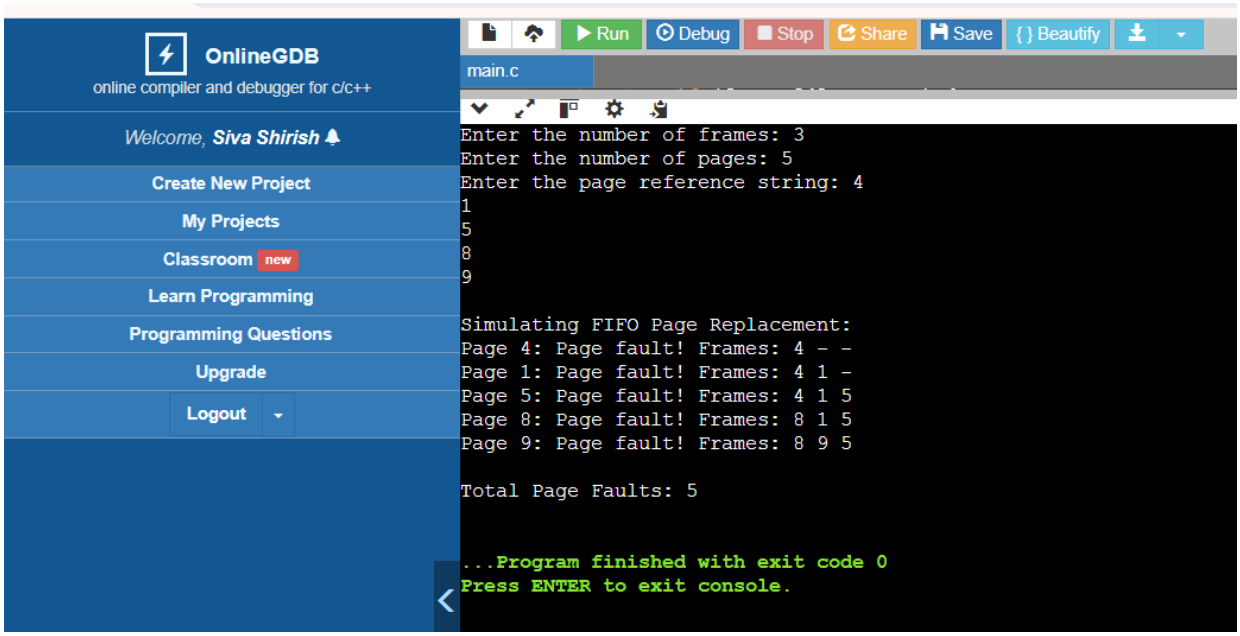
CODE:

```
#include <stdio.h>

#define MAX_FRAMES 10   =
#define MAX_PAGES 50

int main() {
   int frames[MAX_FRAMES], pageRef[MAX_PAGES];
   int numFrames, numPages, pageFaults = 0, front = 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", &pageRef[i]);
   }

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

```
    printf("\nSimulating FIFO Page Replacement:\n");

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

        if (!found)
{
            frames[front] = page;
            front = (front + 1) % numFrames;
            pageFaults++;

            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:


OnlineGDB
online compiler and debugger for c/c++
Welcome, Siva Shirish
Create New Project
My Projects
Classroom new
Learn Programming
Programming Questions
Upgrade
Logout
Run
Debug
Stop
Share
Save
{ } Beautify
main.c
Enter the number of frames: 3
Enter the number of pages: 5
Enter the page reference string: 4
1
5
8
9
Simulating FIFO Page Replacement:
Page 4: Page fault! Frames: 4 - -
Page 1: Page fault! Frames: 4 1 -
Page 5: Page fault! Frames: 4 1 5
Page 8: Page fault! Frames: 8 1 5
Page 9: Page fault! Frames: 8 9 5
Total Page Faults: 5
...Program finished with exit code 0
Press ENTER to exit console.