EXP NO: 11A FIFO

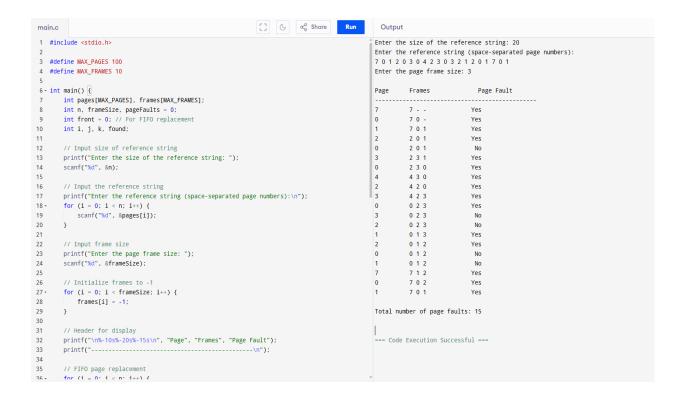
DATE:

CODE:

```
#include <stdio.h>
#define MAX_PAGES 100
#define MAX_FRAMES 10
int main() {
  int pages[MAX_PAGES], frames[MAX_FRAMES];
  int n, frameSize, pageFaults = 0;
  int front = 0; // For FIFO replacement
  int i, j, k, found;
  // Input size of reference string
  printf("Enter the size of the reference string: ");
  scanf("%d", &n);
  // Input the reference string
  printf("Enter the reference string (space-separated page numbers):\n");
  for (i = 0; i < n; i++)
    scanf("%d", &pages[i]);
  }
  // Input frame size
  printf("Enter the page frame size: ");
  scanf("%d", &frameSize);
  // Initialize frames to -1
  for (i = 0; i < frameSize; i++) {
    frames[i] = -1;
  }
  // Header for display
  printf("\n%-10s%-20s%-15s\n", "Page", "Frames", "Page Fault");
  printf("-----\n");
  // FIFO page replacement
  for (i = 0; i < n; i++)
    found = 0;
```

```
// Check if page is already in frame
  for (j = 0; j < \text{frameSize}; j++)
     if (frames[j] == pages[i]) {
       found = 1;
       break;
     }
  }
  // Display current page
  printf("%-10d", pages[i]);
  // If not found, it's a page fault
  if (!found) {
     frames[front] = pages[i];
     front = (front + 1) % frameSize;
     pageFaults++;
  }
  // Display current frame content
  for (k = 0; k < frameSize; k++) {
     if (frames[k] != -1)
       printf("%d ", frames[k]);
     else
       printf("- ");
  }
  // Indicate page fault
  if (!found)
     printf("%15s\n", "Yes");
  else
     printf("%15s\n", "No");
}
// Total page faults
printf("\nTotal number of page faults: %d\n", pageFaults);
return 0;
```

OUTPUT:



RESULT:

Thus the program is executed successfully.

