PAGE REPLACEMENT ALGORITHM (PROG NO – 5)

Objective : Design, develop and implement a C/C++/Java program to implement page replacement algorithm using FIFO. Assume suitable input required to demonstrate the results.

ALGORITHM:

1. **Input:** Total number of pages (**n**), page sequence array (**pg**), page frame size (**psize**).
2. Initialize an array to represent the page frame (**fr**) with a size of **psize** and fill it with -1 to indicate empty slots.
3. Initialize variables **f** (front), **r** (rear), **s** (sequence index), and **count** (page fault count) to 0.
4. While processing pages (**s** less than **n**): a. Set **flag** to 0 to indicate a page fault. b. Get the page number (**num**) from the sequence at index **s**. c. Check if the page is already in the page frame by iterating through the page frame array (**fr**).
    - If found (**flag** is set to 1), increment **s** and continue to the next iteration. d. If the page is not in the page frame:
    - Check if there is space in the page frame (**r** < **psize**):
        - If space is available, add the page to the page frame at the rear (**fr[r] = pg[s]**), and increment both **r** and **s**.
        - Increment the page fault count (**count**).
    - If the page frame is full:
        - Check if there is space at the front (**f** < **psize**):
            - If space is available, add the page to the page frame at the front (**fr[f] = pg[s]**), and increment both **f** and **s**.
            - Increment the page fault count (**count**).
        - If both front and rear are at the end, wrap around to the beginning (**f = 0**). e. Print the current state of the page frame after each page operation.

5. Print the total number of page faults.

PROGRAM:

```
#include <stdio.h>

int n, pg[30], fr[10];

void fifo();


int main()

{
   int i;

   printf("Enter total number of pages: ");
   scanf("%d", &n);

   printf("Enter page sequence:\n");
   for (i = 0; i < n; i++)
      scanf("%d", &pg[i]);

   fifo();

   return 0;
}

void fifo()
{
   int i, f = 0, r = 0, s = 0, count = 0, flag = 0, num, psize;

   printf("Enter the size of page frame: ");
   scanf("%d", &psize);

   for (i = 0; i < psize; i++)
      fr[i] = -1;
```

```
    while (s < n)
 {
        flag = 0;
        num = pg[s];

        for (i = 0; i < psize; i++)
 {
            if (num == fr[i])
           {
               s++;
               flag = 1;
               break;
            }
  }

        if (flag == 0)
{
            if (r < psize)
         {
               fr[r] = pg[s];
               r++;
               s++;
               count++;
            }
else
           {
               if (f < psize)
                 {
                   fr[f] = pg[s];
                   s++;
                   f++;
                   count++;
                 }
 else {
                   f = 0;
               }
```

```
      }
  }

        printf("\nPage Frame: ");
        for (i = 0; i < psize; i++)
            printf("%d ", fr[i]);
    }

    printf("\nPage Faults: %d\n", count);
}
```

Example Input:

```
Enter total number of pages: 12
Enter page sequence:
4 3 2 1 4 3 5 4 3 2 1 5
Enter the size of page frame: 3
```

Example Output:

```
Page Frame: 4 -1 -1
Page Frame: 4 3 -1
Page Frame: 4 3 2
Page Frame: 1 3 2
Page Frame: 1 4 2
Page Frame: 1 4 3
Page Frame: 5 4 3
Page Frame: 5 4 1
Page Frame: 5 4 2
Page Frame: 1 4 2
Page Frame: 1 5 2
Page Frame: 1 5 4
Page Faults: 10
```