

Exp – 11a)

AIM: To find out the number of page faults that occur using First-in First-out (FIFO) page replacement technique.

PROGRAM:

```
#include <stdio.h>

int main() {
    int ref_str[50], frames[10];
    int ref_len, frame_size;
    int i, j, k, page_faults = 0, index = 0, isHit;

    printf("Enter the size of reference string: ");
    scanf("%d", &ref_len);

    for (i = 0; i < ref_len; i++) {
        printf("Enter [ %d ]: ", i + 1);
        scanf("%d", &ref_str[i]);
    }

    printf("\nEnter page frame size : ");
    scanf("%d", &frame_size);

    for (i = 0; i < frame_size; i++)
        frames[i] = -1;

    printf("\n");

    for (i = 0; i < ref_len; i++) {
        isHit = 0;

        // Check if page is already in memory
        for (j = 0; j < frame_size; j++) {
            if (frames[j] == ref_str[i]) {
                isHit = 1;
                break;
            }
        }

        if (!isHit) {
            // FIFO Replacement
            frames[index] = ref_str[i];
            index = (index + 1) % frame_size;
            page_faults++;
        }

        printf("%d -> ", ref_str[i]);
        for (k = 0; k < frame_size; k++) {
            if (frames[k] != -1)
                printf("%d ", frames[k]);
            else
                printf("- ");
        }

        if (isHit)
            printf("-> No Page Fault\n");
        else
            printf("\n");
    }

    printf("\nTotal Page Faults: %d\n", page_faults);

    return 0;
}
```

OUTPUT:

```
jagadesh@LAPTOP-33VRBQ67:/mnt/c/Users/Parthiban/OS Exps/shell/C programs$ ./fifo
Enter the size of reference string: 5
Enter [ 1 ]: 7
Enter [ 2 ]: 0
Enter [ 3 ]: 1
Enter [ 4 ]: 2
Enter [ 5 ]: 0

Enter page frame size : 3

7 -> 7 - -
0 -> 7 0 -
1 -> 7 0 1
2 -> 2 0 1
0 -> 2 0 1 -> No Page Fault

Total Page Faults: 4
```

Exp – 11b)

AIM: To write a C program to implement LRU page replacement algorithm

PROGRAM:

```
#include <stdio.h>

int main() {
    int frames, pages, page_faults = 0;
    int ref_string[100], frame[10], counter[10];
    int i, j, k, pos, max, flag1, flag2, time = 0;

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

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

    printf("Enter reference string: ");
    for(i = 0; i < pages; ++i)
        scanf("%d", &ref_string[i]);

    for(i = 0; i < frames; ++i) {
        frame[i] = -1; // -1 indicates an empty frame
        counter[i] = 0;
    }

    for(i = 0; i < pages; ++i) {
        flag1 = flag2 = 0;

        for(j = 0; j < frames; ++j) {
            if(frame[j] == ref_string[i]) {
                time++;
                counter[j] = time; // Update usage time
                flag1 = flag2 = 1;
                break;
            }
        }

        if(flag1 == 0) {
            for(j = 0; j < frames; ++j) {
                if(frame[j] == -1) {
                    time++;
                    frame[j] = ref_string[i];
                    counter[j] = time;
                    flag2 = 1;
                    page_faults++;
                    break;
                }
            }
        }
    }
}
```

```
        if(flag2 == 0) {
            int min = counter[0];
            pos = 0;
            for(j = 1; j < frames; ++j) {
                if(counter[j] < min) {
                    min = counter[j];
                    pos = j;
                }
            }
            time++;
            frame[pos] = ref_string[i];
            counter[pos] = time;
            page_faults++;
        }

        // Display current frame contents
        for(j = 0; j < frames; ++j)
            printf("%d ", frame[j]);
        printf("\n");
    }

    printf("Total Page Faults = %d\n", page_faults);
    return 0;
}
```

OUTPUT:

```
jagadesh@LAPTOP-33VRBQ67:/mnt/c/Users/Parthiban/OS Exps/shell/C programs$ ./lru
Enter number of frames: 3
Enter number of pages: 5
Enter reference string: 4 6 8 7 2
4 -1 -1
4 6 -1
4 6 8
7 6 8
7 2 8
Total Page Faults = 5
```

Exp – 11c)

AIM: To write a C program to implement Optimal page replacement algorithm

PROGRAM:

```
#include <stdio.h>

int findOptimal(int pages[], int frame[], int currentIndex, int n, int frames) {
    int i, j, index[10], found = 0;

    for (i = 0; i < frames; i++) {
        index[i] = -1;
        for (j = currentIndex + 1; j < n; j++) {
            if (frame[i] == pages[j]) {
                index[i] = j;
                break;
            }
        }
    }

    int farthest = -1, pos = -1;
    for (i = 0; i < frames; i++) {
        if (index[i] == -1) // Not found in future, so replace this
            return i;
        if (index[i] > farthest) {
            farthest = index[i];
            pos = i;
        }
    }
    return pos;
}

int main() {
    int pages[100], frame[10];
    int n, frames, i, j, k, pageFaults = 0, flag, pos;

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

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

    printf("Enter reference string: ");
    for (i = 0; i < n; i++)
        scanf("%d", &pages[i]);

    for (i = 0; i < frames; i++)
        frame[i] = -1;
```

```

for (i = 0; i < n; i++) {
    flag = 0;

    // Check if page is already in frame
    for (j = 0; j < frames; j++) {
        if (frame[j] == pages[i]) {
            flag = 1;
            break;
        }
    }

    if (!flag) {
        if (i < frames) {
            frame[i] = pages[i];
        } else {
            pos = findOptimal(pages, frame, i, n, frames);
            frame[pos] = pages[i];
        }
        pageFaults++;
    }

    // Display current state of frame
    for (k = 0; k < frames; k++) {
        if (frame[k] != -1)
            printf("%d ", frame[k]);
        else
            printf("- ");
    }
    printf("\n");
}

printf("Total Page Faults = %d\n", pageFaults);
return 0;
}

```

OUTPUT:

```

jagadesh@LAPTOP-33VRBQ67:/mnt/c/Users/Parthiban/OS Exps/shell/C programs$ ./optimal
Enter number of frames: 3
Enter number of pages: 5
Enter reference string: 4 5 6 7 8
4 - -
4 5 -
4 5 6
7 5 6
8 5 6
Total Page Faults = 5

```