Exp: 11a) To find out the number of page faults that occur using First-in First-out (FIFO) page replacement technique.

Program code:

```c
#include <stdio.h>

int main() {
    int frames, pages[100], pageFaults = 0;
    int memory[10], front = 0;
    int n, found, i, j;

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

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

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

    for (i = 0; i < frames; i++)
        memory[i] = -1;  // initialize memory with -1 (empty)

    printf("\nPage\tFrames\t\tPage Fault\n");
    for (i = 0; i < n; i++) {
        found = 0;

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

        // If not found, replace the oldest page (FIFO)
        if (!found) {
            memory[front] = pages[i];
            front = (front + 1) % frames;
            pageFaults++;
        }

        // Output current memory
        printf("%d\t", pages[i]);
        for (j = 0; j < frames; j++) {
            if (memory[j] != -1)
                printf("%d ", memory[j]);
            else
                printf("- ");
        }
        printf("\t\t%s\n", found ? "No" : "Yes");
    }

    printf("\nTotal Page Faults = %d\n", pageFaults);

    return 0;
}
```

Output:

```
Enter number of pages: 12
Enter the page reference string:
1 2 3 4 1 2 5 1 2 3 4 5
Enter number of frames: 3

Page    Frames          Page Fault
1       1 - -           Yes
2       1 2 -           Yes
3       1 2 3           Yes
4       4 2 3           Yes
1       4 1 3           Yes
2       4 1 2           Yes
5       5 1 2           Yes
1       5 1 2           No
2       5 1 2           No
3       5 3 2           Yes
4       5 3 4           Yes
5       5 3 4           No

Total Page Faults = 9
```

Exp: 11b)

Aim: To write a c program to implement LRU page replacement algorithm.

Program Code:

```c
#include <stdio.h>

int findLRU(int time[], int n) {
    int i, min = time[0], pos = 0;
    for (i = 1; i < n; ++i) {
        if (time[i] < min) {
            min = time[i];
            pos = i;
        }
    }
    return pos;
}

int main() {
    int frames, pages[100], time[10];
    int counter = 0, faults = 0;
    int i, j, n, flag1, flag2, pos;
    int temp[10];

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

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

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

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

    printf("\nPage\tFrames\t\tPage Fault\n");

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

        // Check if page already exists
        for (j = 0; j < frames; j++) {
            if (temp[j] == pages[i]) {
                counter++;
                time[j] = counter;
                flag1 = flag2 = 1;
                break;
            }
        }

        // If page not found, check for empty slot
        if (!flag1) {
            for (j = 0; j < frames; j++) {
                if (temp[j] == -1) {
                    counter++;
                    faults++;
                    temp[j] = pages[i];
                    time[j] = counter;
                    flag2 = 1;
                    break;
                }
            }
        }

        // If no empty slot, replace LRU page
        if (!flag2) {
            pos = findLRU(time, frames);
            counter++;
            faults++;
            temp[pos] = pages[i];
            time[pos] = counter;
        }

        // Print frame content
        printf("%d\t", pages[i]);
        for (j = 0; j < frames; j++) {
            if (temp[j] != -1)
                printf("%d ", temp[j]);
            else
                printf("- ");
        }
        printf("\t\t%s\n", (flag1 ? "No" : "Yes"));
    }

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

Output:

```
Enter number of pages: 12
Enter page reference string:
1 2 3 4 1 2 5 1 2 3 4 5
Enter number of frames: 3

Page    Frames          Page Fault
1       1 - -           Yes
2       1 2 -           Yes
3       1 2 3           Yes
4       4 2 3           Yes
1       4 1 3           Yes
2       4 1 2           Yes
5       5 1 2           Yes
1       5 1 2           No
2       5 1 2           No
3       3 1 2           Yes
4       3 4 2           Yes
5       3 4 5           Yes

Total Page Faults = 10
```