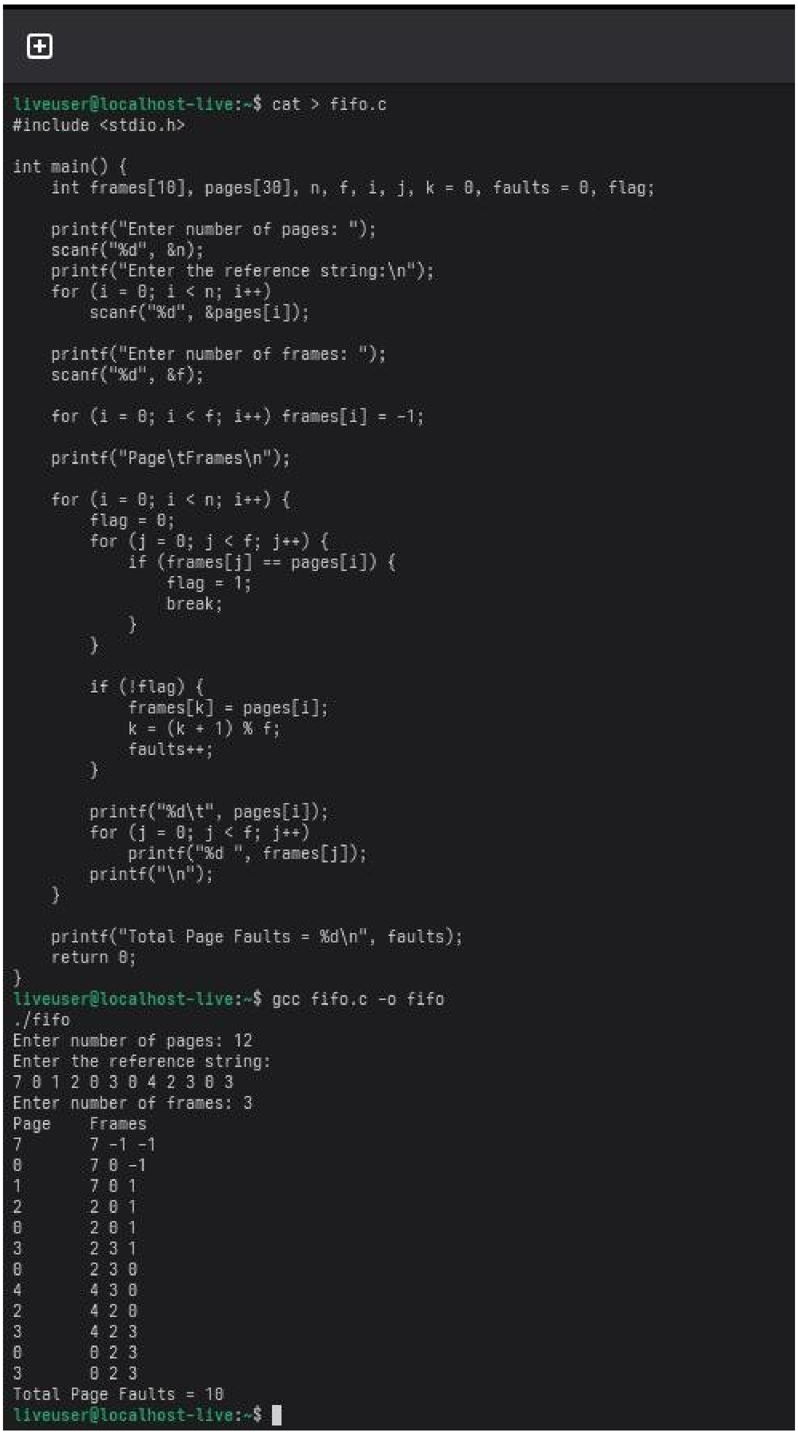
FIFO Page Replacement

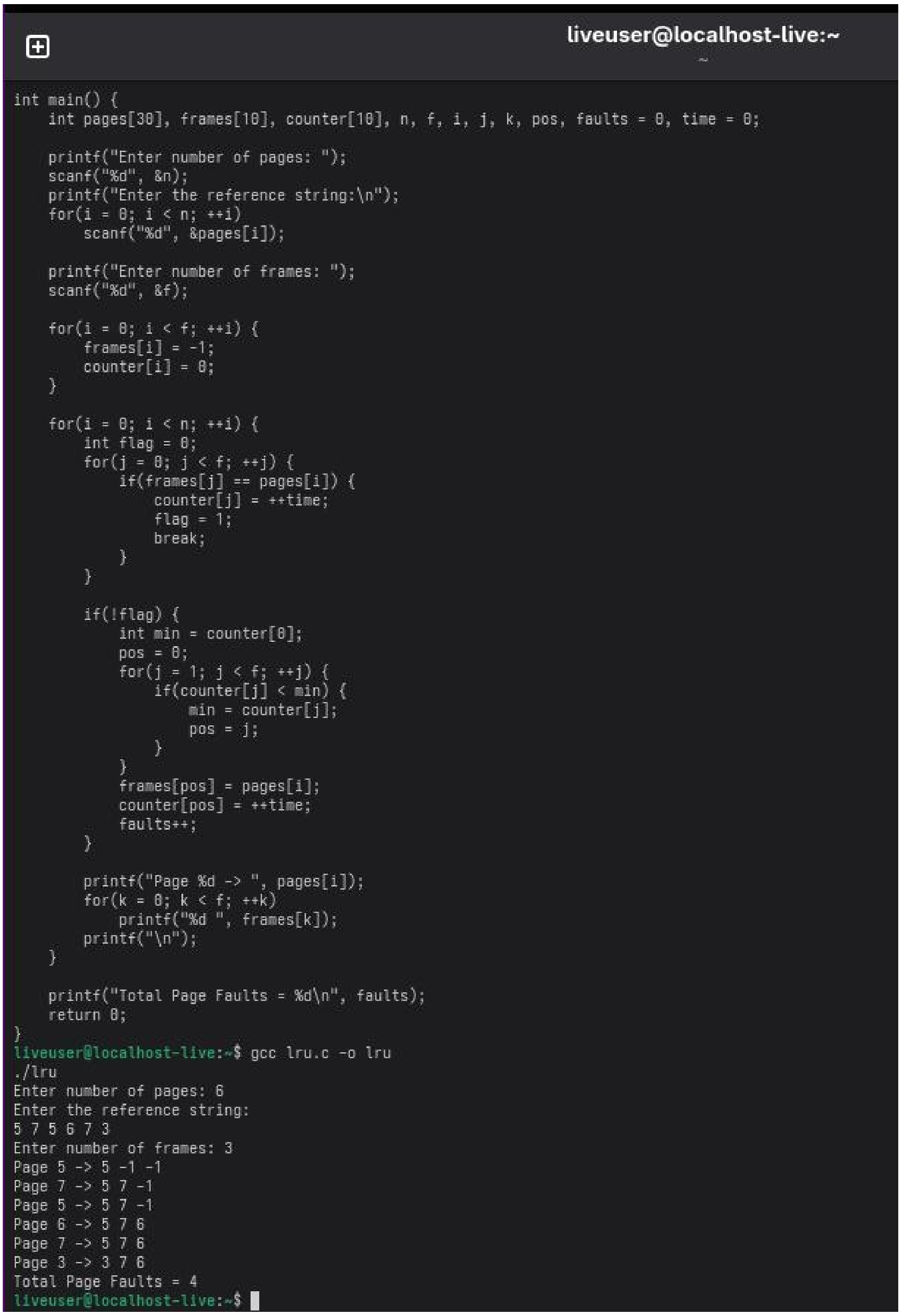
AADITYA P

230701001



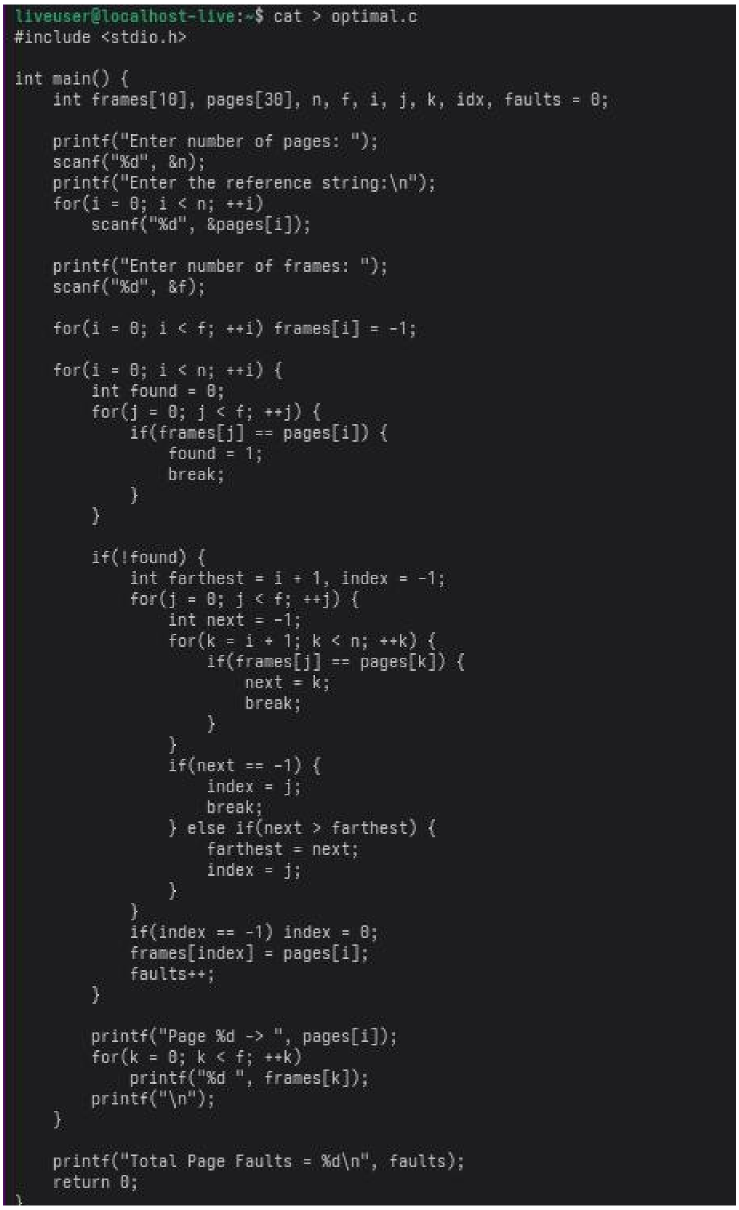
```
liveuser@localhost-live:~$ cat > fifo.c
#include <stdio.h>

int main() {
    int frames[10], pages[30], n, f, i, j, k = 0, faults = 0, flag;

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

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

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

    printf("Page\tFrames\n");

    for (i = 0; i < n; i++) {
        flag = 0;
        for (j = 0; j < f; j++) {
            if (frames[j] == pages[i]) {
                flag = 1;
                break;
            }
        }

        if (!flag) {
            frames[k] = pages[i];
            k = (k + 1) % f;
            faults++;
        }

        printf("%d\t", pages[i]);
        for (j = 0; j < f; j++)
            printf("%d ", frames[j]);
        printf("\n");
    }

    printf("Total Page Faults = %d\n", faults);
    return 0;
}
liveuser@localhost-live:~$ gcc fifo.c -o fifo
./fifo
Enter number of pages: 12
Enter the reference string:
7 0 1 2 0 3 0 4 2 3 0 3
Enter number of frames: 3
Page    Frames
7       7 -1 -1
0       7 0 -1
1       7 0 1
2       2 0 1
0       2 0 1
3       2 3 1
0       2 3 0
4       4 3 0
2       4 2 0
3       4 2 3
0       0 2 3
3       0 2 3
Total Page Faults = 10
liveuser@localhost-live:~$ 
```

LRU Page Replacement



```
int main() {
    int pages[30], frames[10], counter[10], n, f, i, j, k, pos, faults = 0, time = 0;

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

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

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

    for(i = 0; i < n; ++i) {
        int flag = 0;
        for(j = 0; j < f; ++j) {
            if(frames[j] == pages[i]) {
                counter[j] = ++time;
                flag = 1;
                break;
            }
        }

        if(!flag) {
            int min = counter[0];
            pos = 0;
            for(j = 1; j < f; ++j) {
                if(counter[j] < min) {
                    min = counter[j];
                    pos = j;
                }
            }
            frames[pos] = pages[i];
            counter[pos] = ++time;
            faults++;
        }

        printf("Page %d -> ", pages[i]);
        for(k = 0; k < f; ++k)
            printf("%d ", frames[k]);
        printf("\n");
    }

    printf("Total Page Faults = %d\n", faults);
    return 0;
}
liveuser@localhost-live:~$ gcc lru.c -o lru
./lru
Enter number of pages: 6
Enter the reference string:
5 7 5 6 7 3
Enter number of frames: 3
Page 5 -> 5 -1 -1
Page 7 -> 5 7 -1
Page 5 -> 5 7 -1
Page 6 -> 5 7 6
Page 7 -> 5 7 6
Page 3 -> 3 7 6
Total Page Faults = 4
liveuser@localhost-live:~$
```

Optimal Page Replacement



```
liveuser@localhost-live:~$ cat > optimal.c
#include <stdio.h>

int main() {
    int frames[10], pages[30], n, f, i, j, k, idx, faults = 0;

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

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

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

    for(i = 0; i < n; ++i) {
        int found = 0;
        for(j = 0; j < f; ++j) {
            if(frames[j] == pages[i]) {
                found = 1;
                break;
            }
        }

        if(!found) {
            int farthest = i + 1, index = -1;
            for(j = 0; j < f; ++j) {
                int next = -1;
                for(k = i + 1; k < n; ++k) {
                    if(frames[j] == pages[k]) {
                        next = k;
                        break;
                    }
                }
                if(next == -1) {
                    index = j;
                    break;
                } else if(next > farthest) {
                    farthest = next;
                    index = j;
                }
            }
            if(index == -1) index = 0;
            frames[index] = pages[i];
            faults++;
        }

        printf("Page %d -> ", pages[i]);
        for(k = 0; k < f; ++k)
            printf("%d ", frames[k]);
        printf("\n");
    }

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