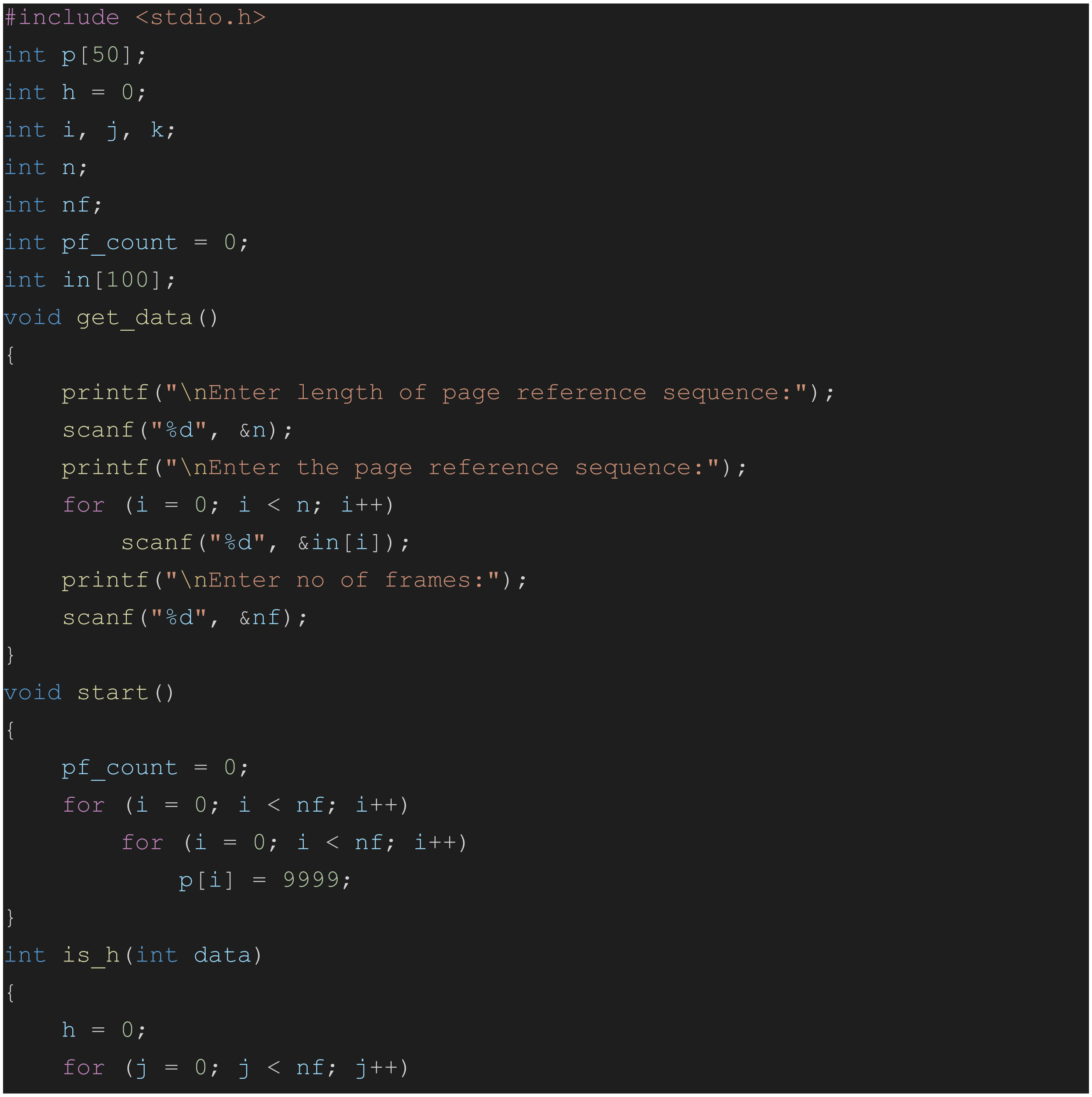
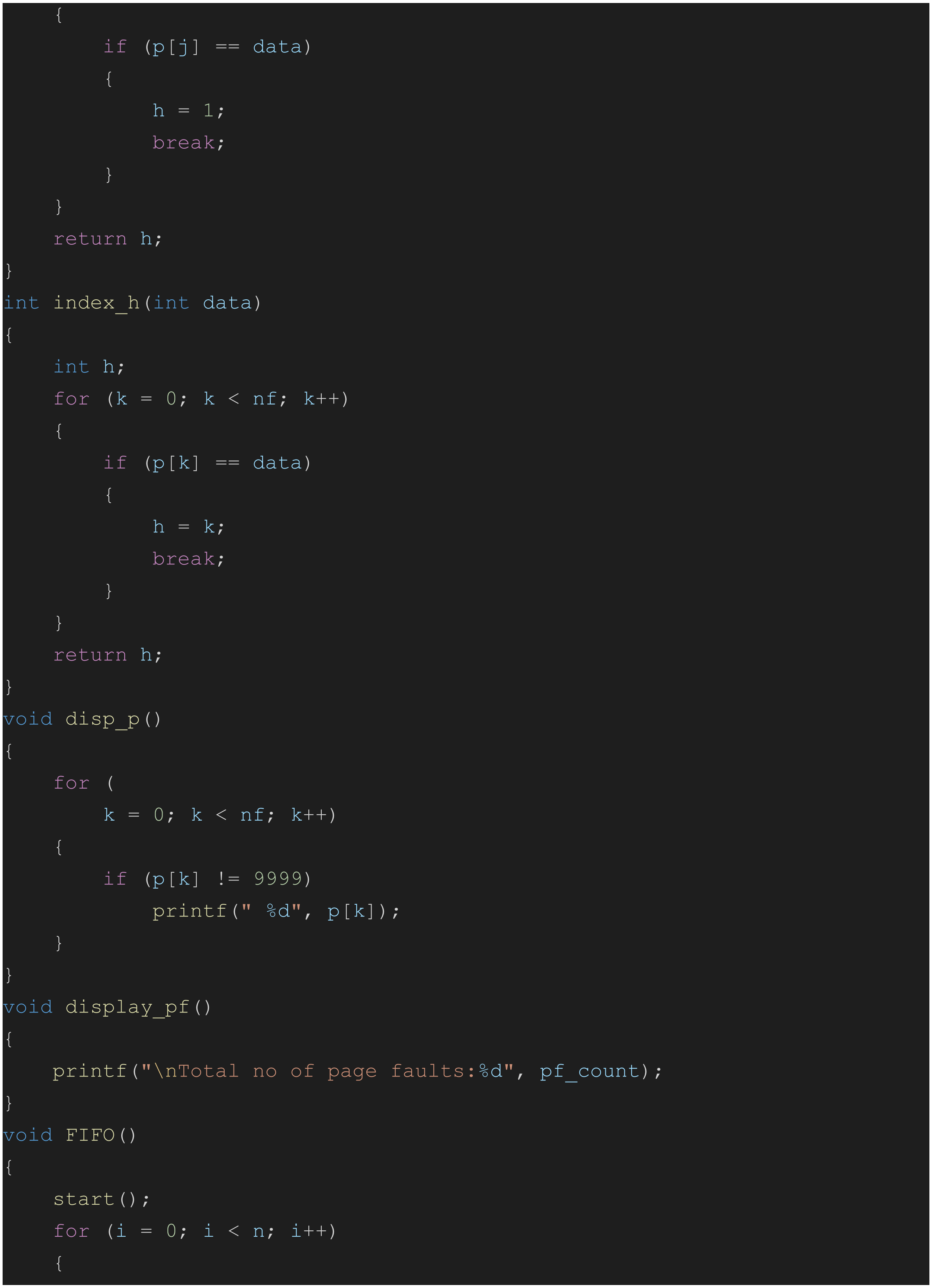
**Name - Kaushal Oza**
**Roll no - 40**
**SRN – 201900754**
**DIV – A**

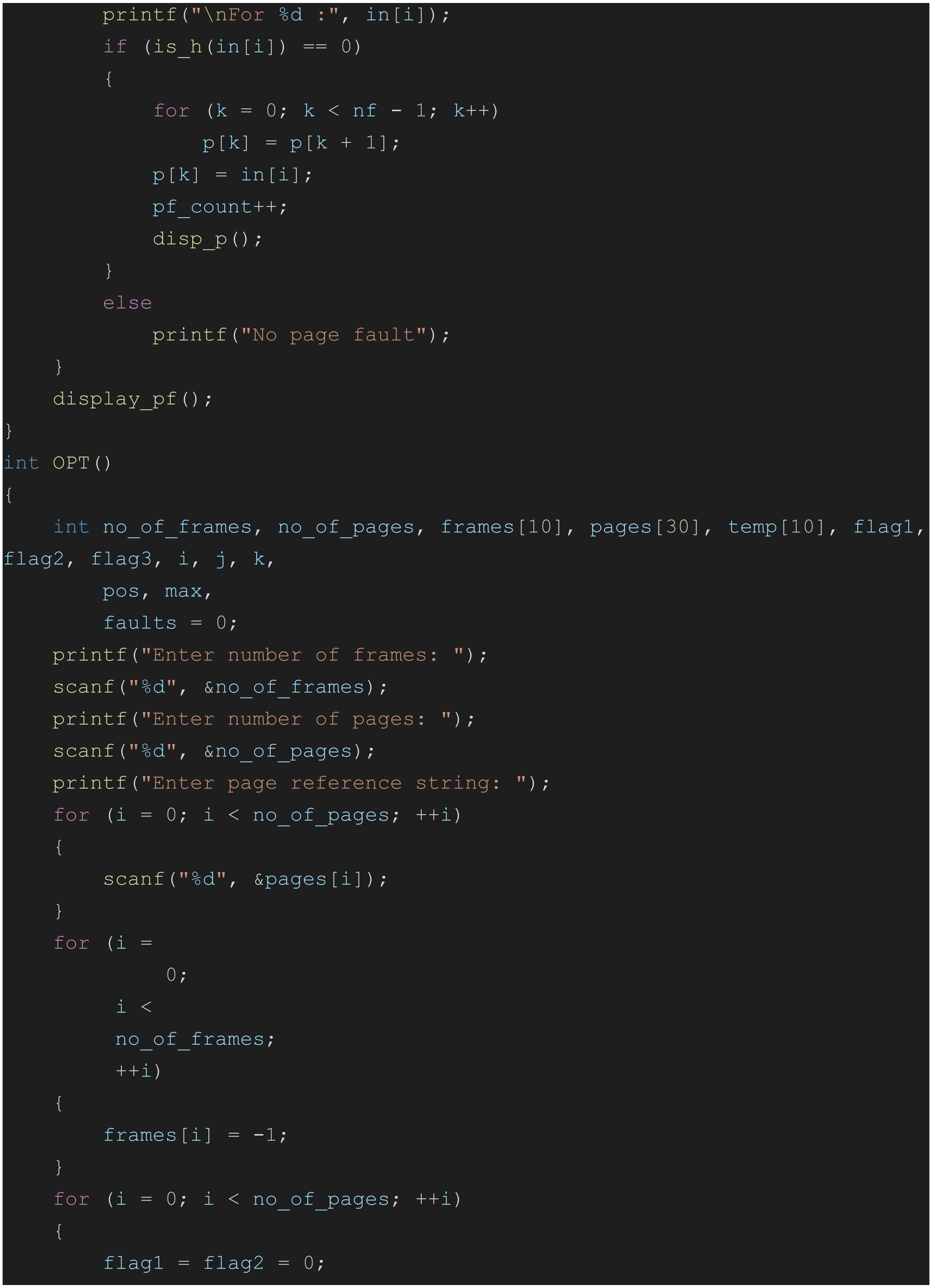
# ♦ OS Assignment -8 ♦



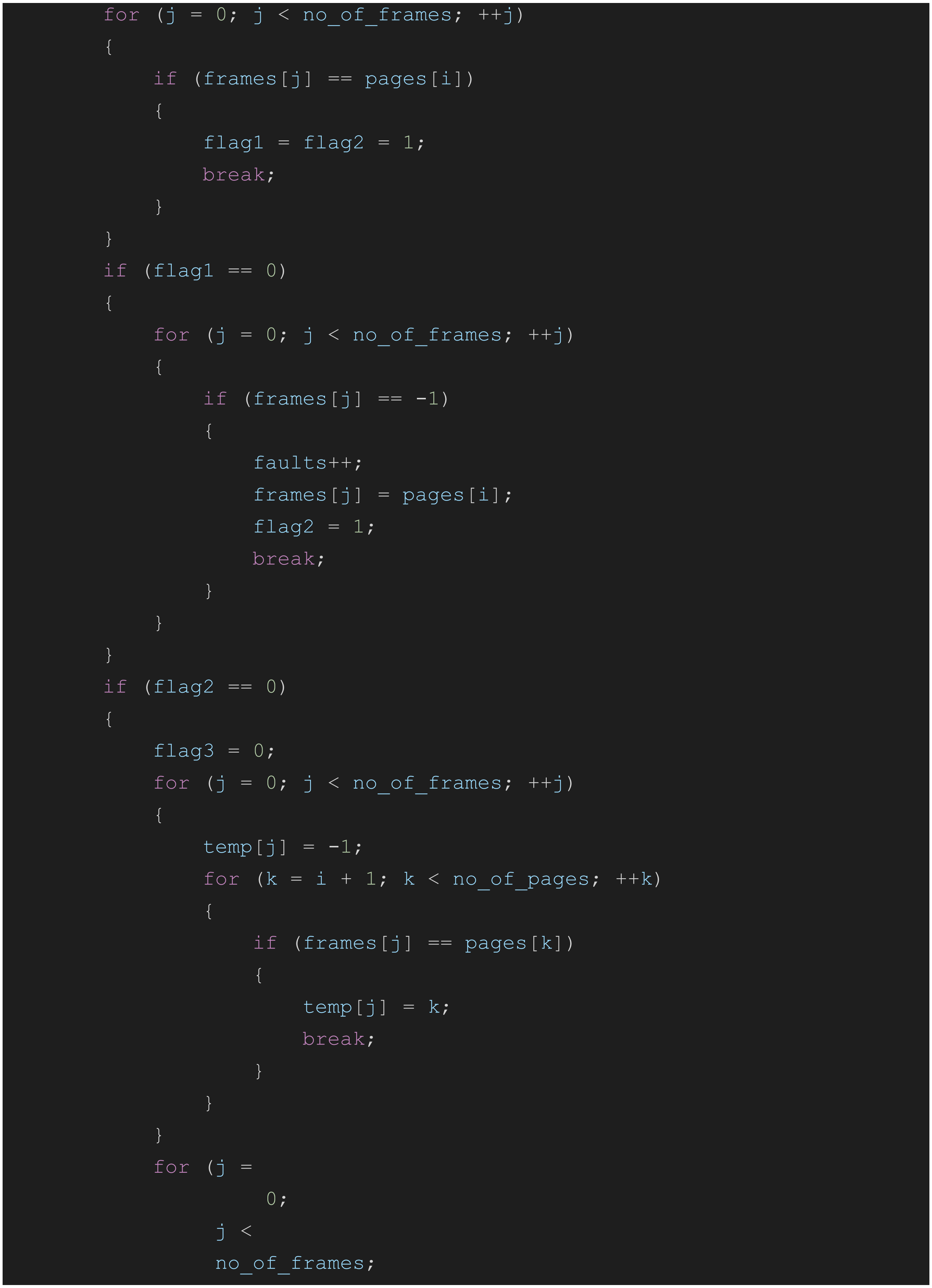
```
#include <stdio.h>
int p[50];
int h = 0;
int i, j, k;
int n;
int nf;
int pf_count = 0;
int in[100];
void get_data()
{
    printf("\nEnter length of page reference sequence:");
    scanf("%d", &n);
    printf("\nEnter the page reference sequence:");
    for (i = 0; i < n; i++)
        scanf("%d", &in[i]);
    printf("\nEnter no of frames:");
    scanf("%d", &nf);
}
void start()
{
    pf_count = 0;
    for (i = 0; i < nf; i++)
        for (i = 0; i < nf; i++)
            p[i] = 9999;
}
int is_h(int data)
{
    h = 0;
    for (j = 0; j < nf; j++)
```



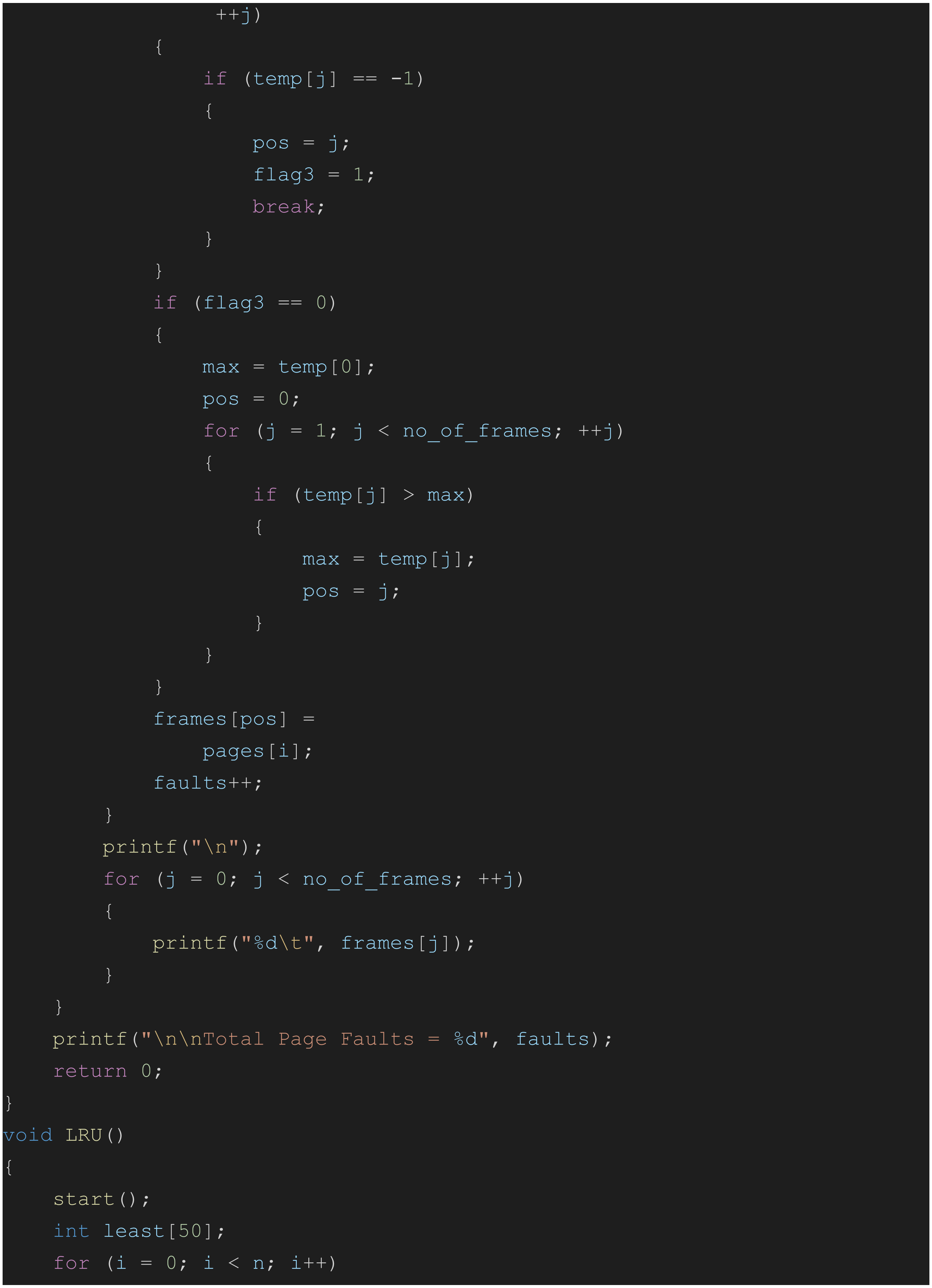
```
    {
        if (p[j] == data)
        {
            h = 1;
            break;
        }
    }
    return h;
}
int index_h(int data)
{
    int h;
    for (k = 0; k < nf; k++)
    {
        if (p[k] == data)
        {
            h = k;
            break;
        }
    }
    return h;
}
void disp_p()
{
    for (
        k = 0; k < nf; k++)
    {
        if (p[k] != 9999)
            printf(" %d", p[k]);
    }
}
void display_pf()
{
    printf("\nTotal no of page faults:%d", pf_count);
}
void FIFO()
{
    start();
    for (i = 0; i < n; i++)
    {
```



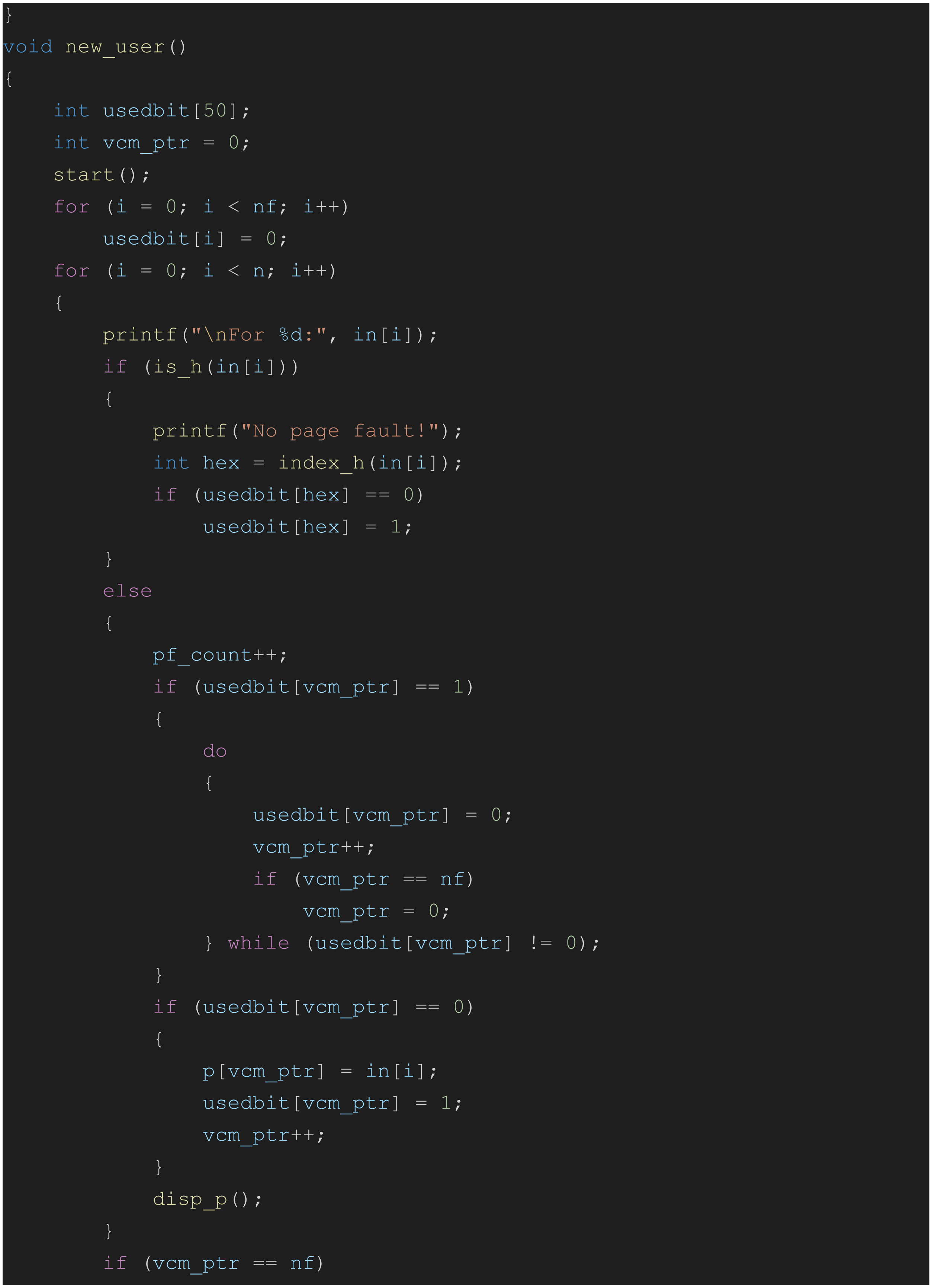
```
        printf("\nFor %d :", in[i]);
        if (is_h(in[i]) == 0)
        {
            for (k = 0; k < nf - 1; k++)
                p[k] = p[k + 1];
            p[k] = in[i];
            pf_count++;
            disp_p();
        }
        else
            printf("No page fault");
    }
    display_pf();
}
int OPT()
{
    int no_of_frames, no_of_pages, frames[10], pages[30], temp[10], flag1,
flag2, flag3, i, j, k,
        pos, max,
        faults = 0;
    printf("Enter number of frames: ");
    scanf("%d", &no_of_frames);
    printf("Enter number of pages: ");
    scanf("%d", &no_of_pages);
    printf("Enter page reference string: ");
    for (i = 0; i < no_of_pages; ++i)
    {
        scanf("%d", &pages[i]);
    }
    for (i =
             0;
         i <
         no_of_frames;
         ++i)
    {
        frames[i] = -1;
    }
    for (i = 0; i < no_of_pages; ++i)
    {
        flag1 = flag2 = 0;
```



```
        for (j = 0; j < no_of_frames; ++j)
        {
            if (frames[j] == pages[i])
            {
                flag1 = flag2 = 1;
                break;
            }
        }
        if (flag1 == 0)
        {
            for (j = 0; j < no_of_frames; ++j)
            {
                if (frames[j] == -1)
                {
                    faults++;
                    frames[j] = pages[i];
                    flag2 = 1;
                    break;
                }
            }
        }
        if (flag2 == 0)
        {
            flag3 = 0;
            for (j = 0; j < no_of_frames; ++j)
            {
                temp[j] = -1;
                for (k = i + 1; k < no_of_pages; ++k)
                {
                    if (frames[j] == pages[k])
                    {
                        temp[j] = k;
                        break;
                    }
                }
            }
            for (j =
                     0;
                 j <
                 no_of_frames;
```



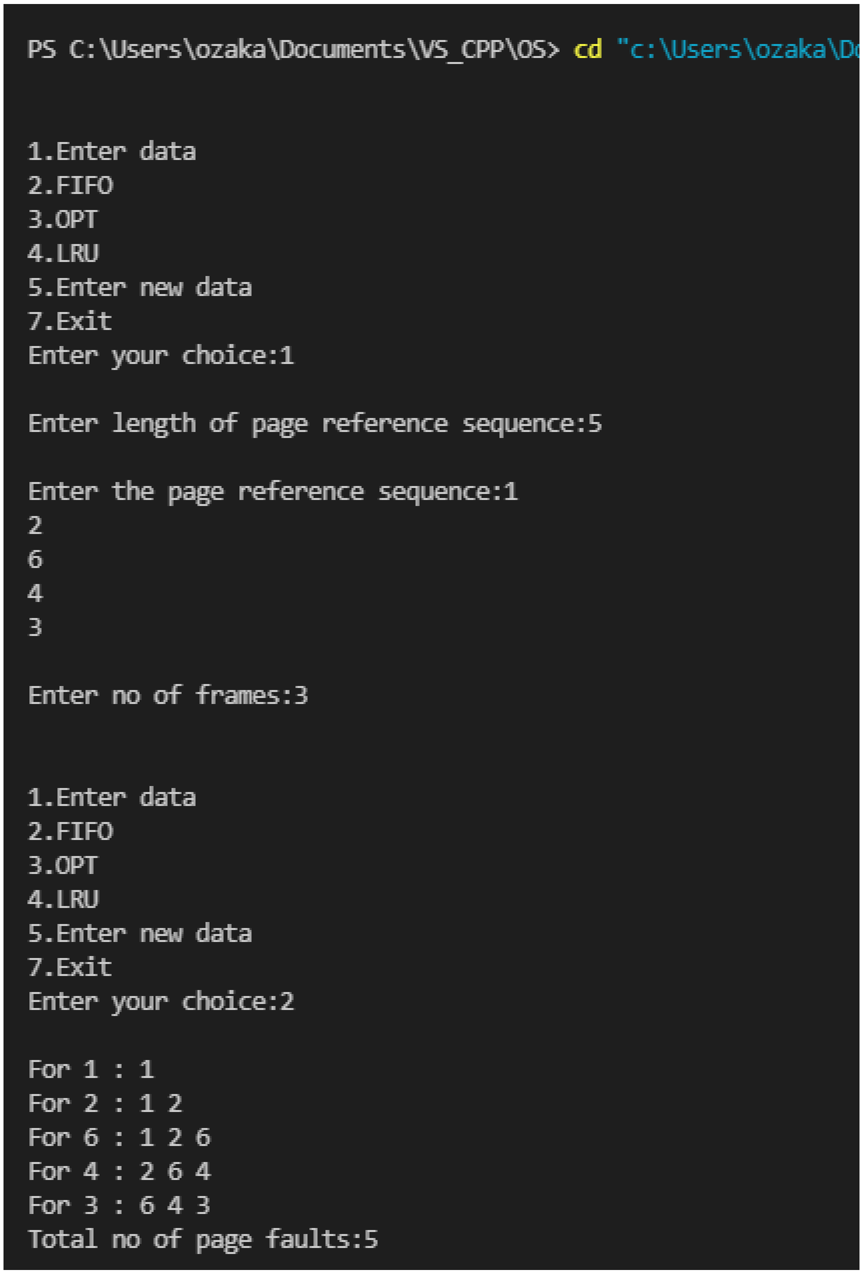
```
                 ++j)
            {
                if (temp[j] == -1)
                {
                    pos = j;
                    flag3 = 1;
                    break;
                }
            }
            if (flag3 == 0)
            {
                max = temp[0];
                pos = 0;
                for (j = 1; j < no_of_frames; ++j)
                {
                    if (temp[j] > max)
                    {
                        max = temp[j];
                        pos = j;
                    }
                }
            }
            frames[pos] =
                pages[i];
            faults++;
        }
        printf("\n");
        for (j = 0; j < no_of_frames; ++j)
        {
            printf("%d\t", frames[j]);
        }
    }
    printf("\n\nTotal Page Faults = %d", faults);
    return 0;
}
void LRU()
{
    start();
    int least[50];
    for (i = 0; i < n; i++)
    {
        printf("\nFor %d :", in[i]);
        if (is_h(in[i]) == 0)
        {
            for (j = 0; j < nf; j++)
            {
                int pg = p[j];
                int found = 0;
                for (k = i - 1; k >= 0; k--)
                {
                    if (pg == in[k])
                    {
                        least[j] = k;
                        found = 1;
                        break;
                    }
                    else
                        found = 0;
                }
                if (!found)
                    least[j] = -9999;
            }
            int min = 9999;
            int repindex;
            for (j = 0; j < nf; j++)
            {
                if (least[j] < min)
                {
                    min = least[j];
                    repindex = j;
                }
            }
            p[repindex] = in[i];
            pf_count++;
            disp_p();
        }
        else
            printf("No page fault!");
    }
    display_pf();
```



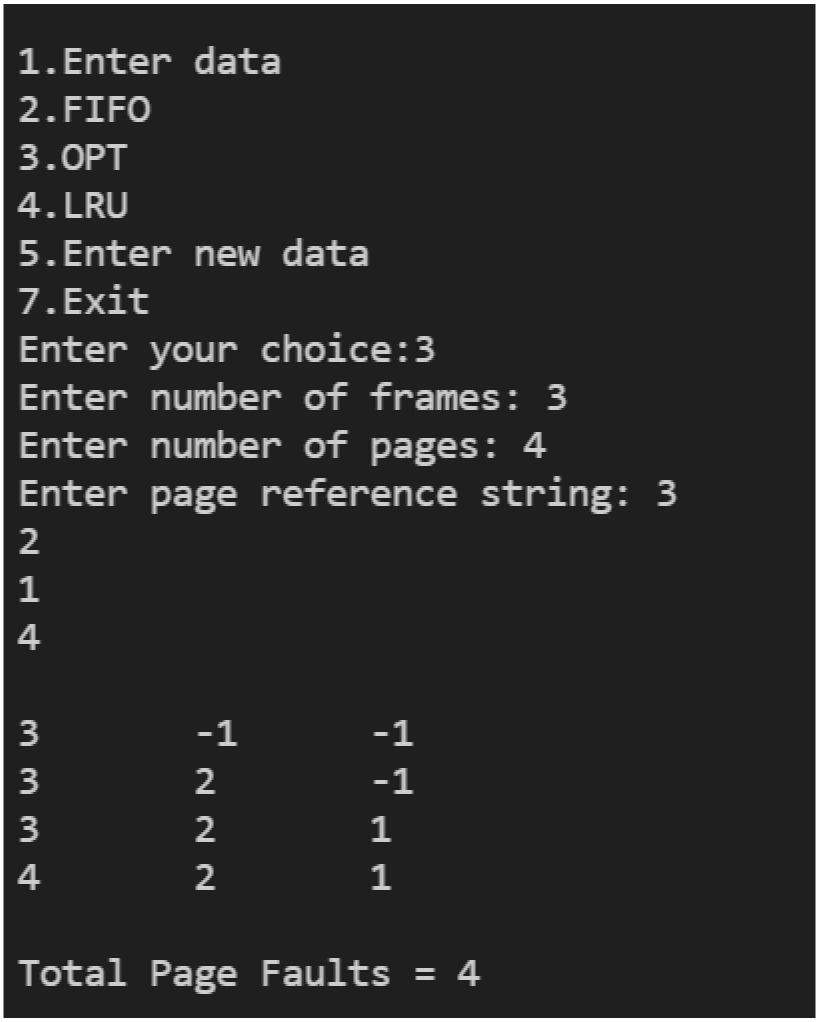
```
}
void new_user()
{
    int usedbit[50];
    int vcm_ptr = 0;
    start();
    for (i = 0; i < nf; i++)
        usedbit[i] = 0;
    for (i = 0; i < n; i++)
    {
        printf("\nFor %d:", in[i]);
        if (is_h(in[i]))
        {
            printf("No page fault!");
            int hex = index_h(in[i]);
            if (usedbit[hex] == 0)
                usedbit[hex] = 1;
        }
        else
        {
            pf_count++;
            if (usedbit[vcm_ptr] == 1)
            {
                do
                {
                    usedbit[vcm_ptr] = 0;
                    vcm_ptr++;
                    if (vcm_ptr == nf)
                        vcm_ptr = 0;
                } while (usedbit[vcm_ptr] != 0);
            }
            if (usedbit[vcm_ptr] == 0)
            {
                p[vcm_ptr] = in[i];
                usedbit[vcm_ptr] = 1;
                vcm_ptr++;
            }
            disp_p();
        }
        if (vcm_ptr == nf)
```

```
            vcm_ptr = 0;
    }
    display_pf();
}
int main()
{
    int choice;
    while (1)
    {
        printf("\n\n1.Enter data\n2.FIFO\n3.OPT\n4.LRU\n5.Enter new data\n7.Exit\nEnter your choice:");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            get_data();
            break;
        case 2:
            FIFO();
            break;
        case 3:
            OPT();
            break;
        case 4:
            LRU();
            break;
        case 5:
            new_user();
            break;
        default:
            return 0;
            break;
        }
    }
}
```

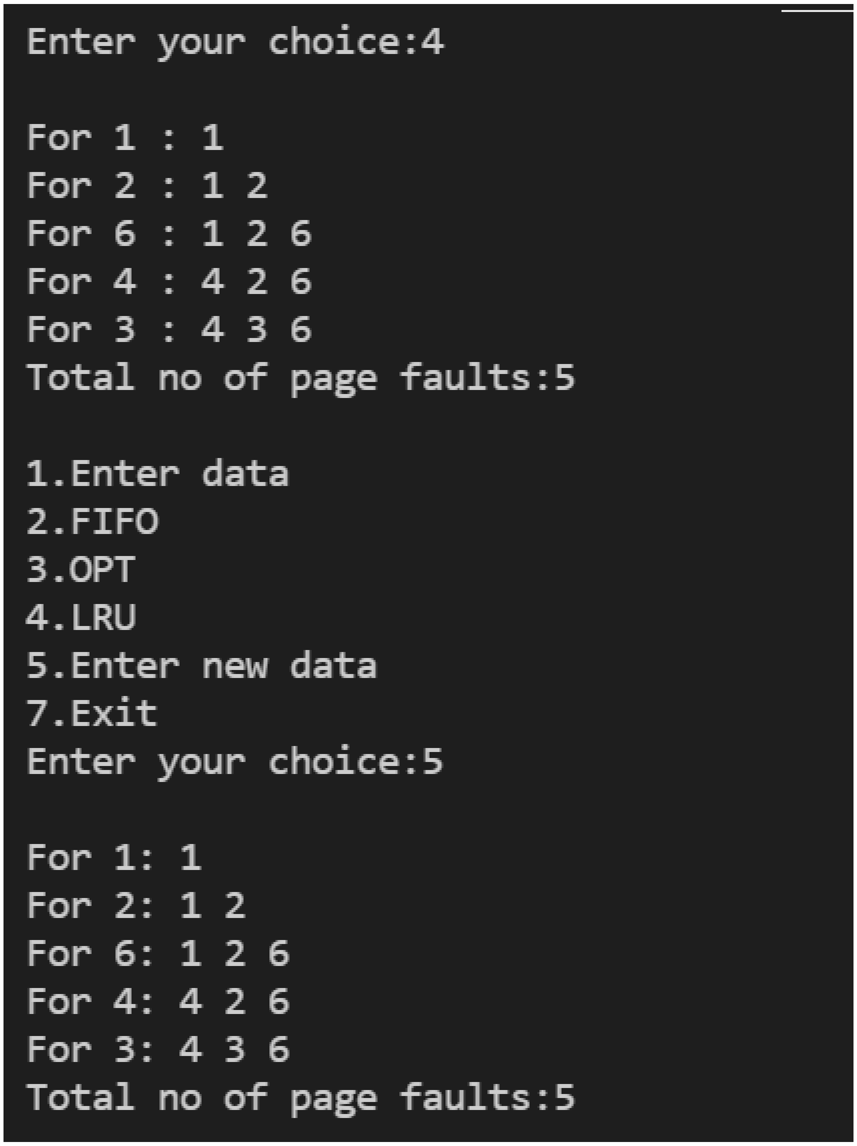
## OUTPUT:



```
PS C:\Users\ozaka\Documents\VS_CPP\OS> cd "c:\Users\ozaka\Do


1.Enter data
2.FIFO
3.OPT
4.LRU
5.Enter new data
7.Exit
Enter your choice:1

Enter length of page reference sequence:5

Enter the page reference sequence:1
2
6
4
3

Enter no of frames:3


1.Enter data
2.FIFO
3.OPT
4.LRU
5.Enter new data
7.Exit
Enter your choice:2

For 1 : 1
For 2 : 1 2
For 6 : 1 2 6
For 4 : 2 6 4
For 3 : 6 4 3
Total no of page faults:5
```



```
1.Enter data
2.FIFO
3.OPT
4.LRU
5.Enter new data
7.Exit
Enter your choice:3
Enter number of frames: 3
Enter number of pages: 4
Enter page reference string: 3
2
1
4

3       -1      -1
3       2       -1
3       2       1
4       2       1

Total Page Faults = 4
```



```
Enter your choice:4

For 1 : 1
For 2 : 1 2
For 6 : 1 2 6
For 4 : 4 2 6
For 3 : 4 3 6
Total no of page faults:5

1.Enter data
2.FIFO
3.OPT
4.LRU
5.Enter new data
7.Exit
Enter your choice:5

For 1: 1
For 2: 1 2
For 6: 1 2 6
For 4: 4 2 6
For 3: 4 3 6
Total no of page faults:5
```