LAB12 HUZAIFA SALMAN DT-34

a) FIFO

CODE

```
#include <stdio.h>
#include <stdlib.h>  // for exit()

int main() {
    int i, j, k, f, pf = 0, count = 0;
    int rs[25], m[10], n;

    // clrscr(); // Not needed in modern compilers

    printf("\nEnter the length of the reference string: ");
    scanf("%d", &n);

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

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

    for (i = 0; i < f; i++) {
        m[i] = -1;  // Initialize all frames to -1
    }

    printf("\nThe Page Replacement Process is:\n");

    for (i = 0; i < n; i++) {
        for (k = 0; k < f; k++) {
            if (m[k] == rs[i]) {
                break;  // Page hit
            }
        }

        if (k == f) {  // Page fault
            m[count++] = rs[i];
            pf++;
        }

        // Display current frame status
        for (j = 0; j < f; j++) {
            if (m[j] != -1)
                printf("\t%d", m[j]);
            else
                printf("\t-");
        }
```

```
        if (k == f)
            printf("\tPF No. %d", pf);

        printf("\n");

        if (count == f)
            count = 0;
    }

    printf("\nTotal Page Faults using FIFO: %d\n", pf);

    // getch(); // Not needed in modern compilers
    return 0;
}
```

```
Enter the length of the reference string: 13
Enter the reference string: 7 0 1 2 0 3 0 4 2 3 0 3 2
Enter the number of frames: 3

The Page Replacement Process is:
        7       -       -       PF No. 1
        7       0       -       PF No. 2
        7       0       1       PF No. 3
        2       0       1       PF No. 4
        2       0       1
        2       3       1       PF No. 5
        2       3       0       PF No. 6
        4       3       0       PF No. 7
        4       2       0       PF No. 8
        4       2       3       PF No. 9
        0       2       3       PF No. 10
        0       2       3
        0       2       3

Total Page Faults using FIFO: 10
```

b) LRU

```
#include <stdio.h>
#include <stdlib.h>  // for exit()

int main() {
    int i, j, k, min, rs[25], m[10], count[10], flag[25];
    int n, f, pf = 0, next = 1;
```

```
    // clrscr(); // Not used in modern compilers

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

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

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

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

    printf("\nThe Page Replacement Process is:\n");

    for (i = 0; i < n; i++) {
        for (j = 0; j < f; j++) {
            if (m[j] == rs[i]) {
                flag[i] = 1;
                count[j] = next++;
                break;
            }
        }

        if (flag[i] == 0) {
            if (i < f) {
                m[i] = rs[i];
                count[i] = next++;
            } else {
                min = 0;
                for (j = 1; j < f; j++) {
                    if (count[min] > count[j]) {
                        min = j;
                    }
                }
                m[min] = rs[i];
                count[min] = next++;
            }
            pf++;
        }

        for (j = 0; j < f; j++) {
            if (m[j] != -1)
```

```
                printf("%d\t", m[j]);
            else
                printf("-\t");
        }

        if (flag[i] == 0)
            printf("PF No. -- %d", pf);
        printf("\n");
    }

    printf("\nTotal number of page faults using LRU: %d\n", pf);

    // getch(); // Not used in modern compilers
    return 0;
}
```

OUTPUT:

```
PS C:\6th-sems\OS labs> cd "c:\6th-sems\OS labs\" ; if ($?)
{ gcc lab_12_2.c -o lab_12_2 } ; if ($?) { .\lab_12_2 }
Enter the length of the reference string: 13
Enter the reference string: 7 0 1 2 0 3 0 4 2 3 0 3 2
Enter the number of frames: 3

The Page Replacement Process is:
7       -       -       PF No. -- 1
7       0       -       PF No. -- 2
7       0       1       PF No. -- 3
2       0       1       PF No. -- 4
2       0       1
2       0       3       PF No. -- 5
2       0       3
4       0       3       PF No. -- 6
4       0       2       PF No. -- 7
4       3       2       PF No. -- 8
0       3       2       PF No. -- 9
0       3       2
0       3       2

Total number of page faults using LRU: 9
```

c) Optimal Page Replacement

```
#include <stdio.h>

int main() {
    int no_of_frames, no_of_pages;
    int frames[10], pages[30], temp[10];
    int flag1, flag2, flag3;
    int 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;

        // Check if page is already in a frame
        for (j = 0; j < no_of_frames; ++j) {
            if (frames[j] == pages[i]) {
                flag1 = flag2 = 1;
                break;
            }
        }

        // If page is not in frame but there is empty space
        if (flag1 == 0) {
            for (j = 0; j < no_of_frames; ++j) {
                if (frames[j] == -1) {
                    faults++;
                    frames[j] = pages[i];
                    flag2 = 1;
                    break;
                }
            }
        }

        // If page is not in frame and no empty space, apply Optimal Replacement
        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++;
    }

    // Print current state of memory frames
    printf("\n");
    for (j = 0; j < no_of_frames; ++j) {
        if (frames[j] != -1)
            printf("%d\t", frames[j]);
        else
            printf("-\t");
    }
}

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

```
PS C:\6th-sems\OS labs> cd "c:\6th-sems\OS labs\" ; if ($?)
{ gcc lab_12_3.c -o lab_12_3 } ; if ($?) { .\lab_12_3 }
Enter number of frames: 3
Enter number of pages: 13
Enter page reference string: 7 0 1 2 0 3 0 4 2 3 0 3 2

7       -       -
```

d) MRU

```
#include <stdio.h>

// Function to update the array in most recently used fashion
void recently(int* arr, int size, int elem)
{
    int index = elem % size;
    int temp = index;
    int id = arr[index];

    // Shift elements from index to 1 position right
    while (temp > 0)
    {
        arr[temp] = arr[temp - 1];
        temp--;
    }

    // Place the element at the front
    arr[0] = id;
}

// Function to print array elements
void print(int* arr, int size)
{
    for (int i = 0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

int main() {
    int elem = 3;
```

```
    int arr[] = { 6, 1, 9, 5, 3 };
    int size = sizeof(arr) / sizeof(arr[0]);

    recently(arr, size, elem);
    printf("Array in most recently used fashion: ");
    print(arr, size);

    return 0;
}
```

```
PS C:\6th-sems\OS labs> cd "c:\6th-sems\OS labs\" ; if ($?)
{ gcc lab_12_4.c -o lab_12_4 } ; if ($?) { .\lab_12_4 }
Array in most recently used fashion: 5 6 1 9 3
PS C:\6th-sems\OS labs>
```