OPERATING SYSTEMS (CT-353) LAB 12

NAME: Ajiya Anwar ROLL NO: DT-22006

CODE:

FIFO:

```
#include <stdio.h>

#include <conio.h>

int main() {

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

   printf("\nEnter the length of reference string: ");

   scanf("%d", &n);

   printf("\nEnter the reference string: ");

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

      scanf("%d", &rs[i]);

   printf("\nEnter number of frames: ");

   scanf("%d", &f);

   for (i = 0; i < f; i++)

      m[i] = -1;  // Initialize all frames to -1 (empty)

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

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

      for (k = 0; k < f; k++) {

         if (m[k] == rs[i]) // Page hit

            break; }

      if (k == f) {  // Page not found, i.e., Page Fault

         m[count++] = rs[i];

         pf++;

      }

      for (j = 0; j < f; j++)

         printf("\t%d", m[j]);

      if (k == f)

         printf("\tPF No. %d", pf);
```

```
    printf("\n");
    if (count == f)  // Reset counter for FIFO
        count = 0; }
  printf("\nTotal number of Page Faults using FIFO: %d", pf);
  getch();
}
```

OUTPUT:

```
C:\Users\Sabri\OneDrive\Desl

Enter the length of reference string: 12

Enter the reference string: 1 2 3 4 1 2 5 1 2 3 4 5

Enter number of frames: 3

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

Total number of Page Faults using FIFO: 9
```

OPERATING SYSTEMS (CT-353) LAB 12

CODE:

LRU:

```
#include <stdio.h>

#include <conio.h>

int main() {

   int i, j, k, min, rs[25], m[10], count[10], flag[25];

   int n, f, pf = 0, next = 1;

   printf("Enter the length of 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++) {

      // Check if the page is already in memory

      for (j = 0; j < f; j++) {

         if (m[j] == rs[i]) {

            flag[i] = 1;       // Page hit

            count[j] = next++;  // Update usage time

         }

      }
```

```
        // If page fault
        if (flag[i] == 0) {
            if (i < f) {
                m[i] = rs[i];
                count[i] = next++;
            } else {
                // Find least recently used page
                min = 0;
                for (j = 1; j < f; j++) {
                    if (count[min] > count[j])
                        min = j;
                }
                m[min] = rs[i];
                count[min] = next++;
            }
            pf++; // Increment page faults
        }
        for (j = 0; j < f; j++)
            printf("%d\t", m[j]);
        if (flag[i] == 0)
            printf("PF No. -- %d", pf);
        printf("\n");
    }
    printf("\nThe number of page faults using LRU are %d", pf);
    getch();
}
```

OPERATING SYSTEMS (CT-353) LAB 12

OUTPUT:

```
C:\Users\Sabri\OneDrive\Desl

Enter the length of reference string -- 12
Enter the reference string -- 1 2 3 4 1 2 5 1 2 3 4 5
Enter the number of frames -- 3

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

The number of page faults using LRU are 10
```

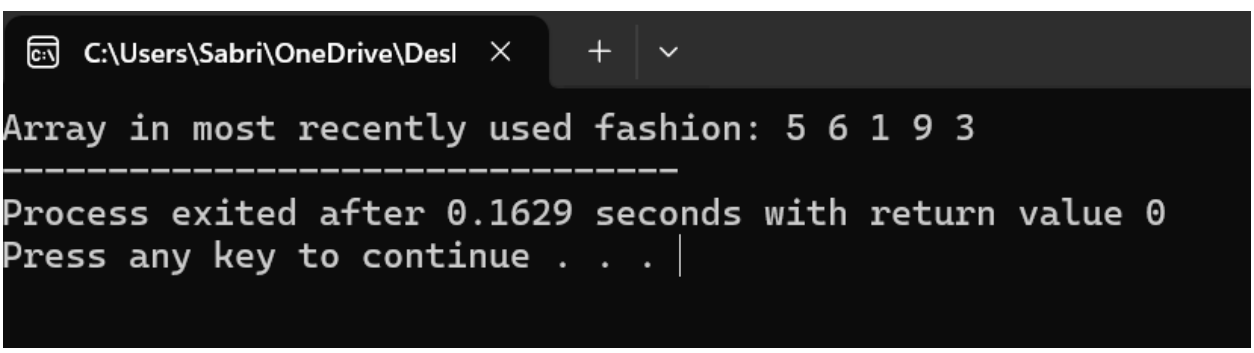
CODE:

MRU:

```
#include <bits/stdc++.h>

using namespace std;

void recently(int* arr, int size, int elem) {

    int index = (elem % size);  // Calculate index based on modulus

    int temp = index;

    int id = arr[index];        // Store the element at that index

    while (temp > 0) {

        arr[temp] = arr[--temp];
```

```
    }
    arr[0] = id;  // Place the selected element at the front
}
void print(int* arr, int size) {
    for (int i = 0; i < size; i++)
        cout << arr[i] << " ";
}
int main() {
    int elem = 3;
    int arr[] = {6, 1, 9, 5, 3};
    int size = sizeof(arr) / sizeof(arr[0]);
    recently(arr, size, elem);
    cout << "Array in most recently used fashion: ";
    print(arr, size);

    return 0;
}
```

OUTPUT:



OPERATING SYSTEMS (CT-353) LAB 12

CODE:

Optimal:

```
#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]);
    }
    // Initialize all frames to -1 (empty)
    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 already exists in frame (Page Hit)
        for (j = 0; j < no_of_frames; ++j) {
            if (frames[j] == pages[i]) {
                flag1 = flag2 = 1;
                break;
```

```
    }
}

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

// If no empty frame, 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++;
}

// Display current state of frames
printf("\n");
for (j = 0; j < no_of_frames; ++j) {
    printf("%d\t", frames[j]);
}
```

```
    }


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

return 0;

}
```

OUTPUT:

```
C:\Users\Sabri\OneDrive\Desl

Enter number of frames: 3
Enter number of pages: 12
Enter page reference string: 7 0 1 2 0 3 0 4 2 3 0 3

7       -1      -1
7       0       -1
7       0       1
2       0       1
2       0       1
2       0       3
2       0       3
2       4       3
2       4       3
2       4       3
0       4       3
0       4       3

Total Page Faults = 7
--------------------------------
Process exited after 47.86 seconds with return value 0
Press any key to continue . . .
```