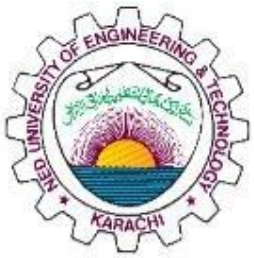

NED UNIVERSITY OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & IT

Specialization in Data Science

CT-353
OPERATING SYSTEMS

Name : Afifa Siddique
Roll No : DT-22003

Submitted to : Sir Muhammad Abdullah Siddiqui

LAB : 12

FIFO

LAB 12 (FIFO).cpp | LAB 12 (LRU)cpp.cpp | LAB 12 (OPTIMAL)cpp.cpp | LAB 12 (MRU).cpp

```
#include <stdio.h>
#include <conio.h>

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

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

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

    printf("\n Enter no. of frames -- ");
    scanf("%d", &f);

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

    printf("\n The Page Replacement Process is -- \n");
    for (i = 0; i < n; i++)
    {
        for (k = 0; k < f; k++)
        {
            if (m[k] == rs[i])
                break;
        }

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

    printf("\n The number of Page Faults using FIFO are %d", pf);
    getch();
```



```
C:\Users\marya\Downloads\O

Enter the length of reference string -- 12

Enter the reference string -- 1 2 3 4 5 6 7 8 9 1 2 3

Enter no. 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       5       3       PF No. 5
        4       5       6       PF No. 6
        7       5       6       PF No. 7
        7       8       6       PF No. 8
        7       8       9       PF No. 9
        1       8       9       PF No. 10
        1       2       9       PF No. 11
        1       2       3       PF No. 12

The number of Page Faults using FIFO are 12
```

LRU

LAB 12 (FIFO).cpp | LAB 12 (LRU)cpp.cpp | LAB 12 (OPTIMAL)cpp.cpp | LAB 12 (MRU).cpp

```c
#include <stdio.h>
#include <conio.h>

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

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

        if (flag[i] == 0)
        {
            if (i < f)
            {
                m[i] = rs[i];
                count[i] = next;
                next++;
            }
            else
```

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

        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();
    return 0;
}
```

```
C:\Users\marya\Downloads\O
Enter the length of reference string -- 12
Enter the reference string -- 1 2 3 4 1 2 3 4 5 6 7 12
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
3       1       2       PF No. -- 7
3       4       2       PF No. -- 8
3       4       5       PF No. -- 9
6       4       5       PF No. -- 10
6       7       5       PF No. -- 11
6       7       12      PF No. -- 12

The number of page faults using LRU are 12
```

OPTIMAL

LAB 12 (FIFO).cpp | LAB 12 (LRU)cpp.cpp | LAB 12 (OPTIMAL)cpp.cpp | LAB 12 (MRU).cpp

```c
#include <stdio.h>

int main()
{
    int no_of_frames, no_of_pages, frames[10], pages[30], temp[10];
    int 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;
                }
            }
        }
```

```
52          if (flag2 == 0)
53          {
54              flag3 = 0;
55
56              for (j = 0; j < no_of_frames; ++j)
57              {
58                  temp[j] = -1;
59
60                  for (k = i + 1; k < no_of_pages; ++k)
61                  {
62                      if (frames[j] == pages[k])
63                      {
64                          temp[j] = k;
65                          break;
66                      }
67                  }
68              }
69
70              for (j = 0; j < no_of_frames; ++j)
71              {
72                  if (temp[j] == -1)
73                  {
74                      pos = j;
75                      flag3 = 1;
76                      break;
77                  }
78              }
79
80              if (flag3 == 0)
81              {
82                  max = temp[0];
83                  pos = 0;
84
85                  for (j = 1; j < no_of_frames; ++j)
86                  {
87                      if (temp[j] > max)
88                      {
89                          max = temp[j];
90                          pos = j;
91                      }
92                  }
93              }
94
95              frames[pos] = pages[i];
96              faults++;
97          }
```

```
94                  }
95
96                  frames[pos] = pages[i];
97                  faults++;
98              }
99
100             printf("\n");
101             for (j = 0; j < no_of_frames; ++j)
102             {
103                 printf("%d\t", frames[j]);
104             }
105         }
106
107         printf("\n\nTotal Page Faults = %d", faults);
108
109         return 0;
110     }
```

```
C:\Users\marya\Downloads\O
Enter number of frames: 10
Enter number of pages: 1 2 3 4 5 6 7 8 9 10
Enter page reference string:
2       -1      -1      -1      -1      -1      -1      -1      -1      -1

Total Page Faults = 1
--------------------------------
Process exited after 12.56 seconds with return value 0
Press any key to continue . . .
```

MRU

LAB 12 (FIFO).cpp | LAB 12 (LRU)cpp.cpp | LAB 12 (OPTIMAL)cpp.cpp | LAB 12 (MRU).cpp

```cpp
#include <bits/stdc++.h>
using namespace std;

// Function to update the array in most recently used fashion
void recently(int* arr, int size, int elem)
{
    int index = 0;
    index = (elem % size);
    int temp = index, id = arr[index];

    while (temp > 0)
    {
        arr[temp] = arr[--temp];
    }
    arr[0] = id;
}

// Print array elements
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;
}

```

```
C:\Users\marya\Downloads\C
array in most recently used fashion : 5 6 1 9 3
--------------------------------
Process exited after 0.217 seconds with return value 0
Press any key to continue . . .
```