Page Replacement-Additional problem

Perumalla Dharan
AP21110010201

(Q) Study about Least Recently Used page replacement policy and implement the same.

```
// Creation Date: 16-05-2023
// LRU Page Replacement Algorithm


// Initialising header files
#include <iostream>
using namespace std;

// Main function
int main(){

    // Taking input from the user
    int n;
    cout << "Enter the number of pages: ";
    cin >> n;

    int pages[n];
    cout << "Enter the pages: ";
    for (int i = 0; i < n; i++)
    {
        cin >> pages[i];
    }

    int cap;
    cout << "Enter the capacity of the frame: ";
    cin >> cap;
```

```
// Initialising the frame
int frame[cap];
for (int i = 0; i < cap; i++)
{
    frame[i] = -1;
}

// Simulating LRU Page Replacement Algorithm
int page_fault = 0;
int index = 0;
int count = 0;
for (int i = 0; i < n; i++)
{
    bool flag = false;
    for (int j = 0; j < cap; j++)
    {
        if (frame[j] == pages[i])
        {
            flag = true;
            break;
        }
    }
    if (!flag)
    {
        if (count < cap)
        {
            frame[index] = pages[i];
            index = (index + 1) % cap;
            count++;
        }
        else
```

```
        {
            int temp[cap];
            for (int j = 0; j < cap; j++)
            {
                temp[j] = -1;
            }
            int k = 0;
            for (int j = i - 1; j >= 0; j--)
            {
                bool flag = false;
                for (int l = 0; l < cap; l++)
                {
                    if (temp[l] == pages[j])
                    {
                        flag = true;
                        break;
                    }
                }
                if (!flag)
                {
                    temp[k] = pages[j];
                    k++;
                }
                if (k == cap)
                {
                    break;
                }
            }
            for (int j = 0; j < cap; j++)
            {
                if (frame[j] == temp[cap - 1])
                {
```

```
                        frame[j] = pages[i];
                        break;
                    }
                }
            }
            page_fault++;
        }
    }

    // Printing the result
    cout << "Number of page faults: " << page_fault <<
endl;
}
```

OUTPUT

```
Enter the number of pages: 8
Enter the pages: 3 5 7 4 1 0 2 5
Enter the capacity of the frame: 8
Number of page faults: 7
PS E:\SRM\OS\OS LAB>
```