Practical No. 7

* Aim:- Implement page replacement algorith

* Theory:-
Page replacement is a crucial aspect of memory management in computer operating system when the physical memory (RAM) is full and a new page needs to be to brought in the O.S must decide which page to remove from memory to make space

- Page replacement:-
Page replacement occurs in virtual memory system when the OS stores data that doesn't fit into physical memory entirely.

- Pages:- Memory is divided into fixed size blocks called pages

- Page table:- Each process has a page tables that maps virtual pages to physical pages.

- Page fault:- When a process access a page not currently in physical memory a page fault occurs triggering page replacement.

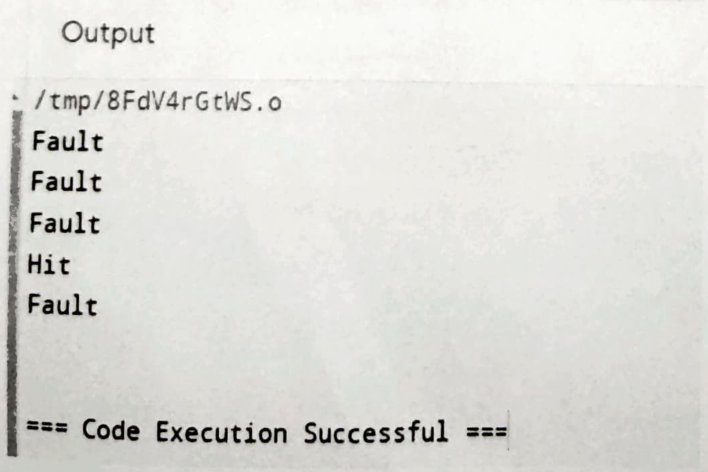
Page replacement is a fundamental aspects of memory management in model. O.S, by understanding the principles and implemented efficient algorithms system designers can optimize memory [illegible] utilization and enhance overall system performance.

* **Result:-**
Hence, we successfully performed & implemented page replacement algorithm.

P	T	D	K	Total
3M	3M	3M	[illegible]	15M
Sign With date				

Practical 7 page replacement

```
#include <iostream>
#include <unordered_set>
#include <queue>
class FIFO {
private:
    int capacity;
    std::queue<int> memory;
    std::unordered_set<int> pageSet;
public:
    FIFO(int capacity) : capacity(capacity) {}
    std::string pageFault(int page) {
        if (pageSet.find(page) == pageSet.end()) {
            if (memory.size() == capacity) {
                int evictedPage = memory.front();
                memory.pop();
                pageSet.erase(evictedPage);
            }
            memory.push(page);
            pageSet.insert(page);
            return "Fault";
        }
        return "Hit";
    }
};
int main() {
    FIFO fifo(3);
    std::cout << fifo.pageFault(1) << std::endl; // Fault
    std::cout << fifo.pageFault(2) << std::endl; // Fault
    std::cout << fifo.pageFault(3) << std::endl; // Fault
    std::cout << fifo.pageFault(1) << std::endl; // Hit
    std::cout << fifo.pageFault(4) << std::endl; // Fault
    return 0;
}
```



Output

```
/tmp/8FdV4rGtWS.o
Fault
Fault
Fault
Hit
Fault


=== Code Execution Successful ===
```