Experiment – 8

NAME : Madhuram Brijeshkumar Modi

ROLL NO. : 21BCP102

DIV, GROUP : 2,G3

Page Replacement Algorithm

1. **First in First Out Page Replacement Algorithm**

```c
#include <stdio.h>
int main(){
    int oldest_page = 0,n,capacity,faults = 0;
    int pages[100];
    printf("Enter the number of pages: ");
    scanf("%d",&n);
    printf("Enter the String of pages: ");
    for (int i=0;i<n;i++){
        scanf("%d",&pages[i]);
    }
    printf("Enter the memory capacity");
    scanf("%d",&capacity);
    int memory[capacity];
    for (int i=0;i<capacity;i++){
        memory[i] = -1;
    }
    for (int i=0;i<n;i++){
        int page = pages[i];
        int page_found = 0;
        for (int j = 0;j<capacity;j++){
            if (page == memory[j]){
                page_found = 1;
                break;
            }
        }
        if (!page_found){
            memory[oldest_page] = page;
            oldest_page = (oldest_page+1)%capacity;
            faults++;
        }
        printf("Memory: ");
        for (int j=0;j<capacity;j++){
            if (memory[j] == -1){
                printf("X ");
            }
            else{
                printf("%d ",memory[j]);
            }
        }
        printf("\n");
    }

    printf("Total Faults = %d\n",faults);
}
```

OUTPUT

```
Enter the number of pages: 10
Enter the String of pages: 7
5 9 4 3 7 9 6 2 1
Enter the memory capacity3
Memory: 7 X X
Memory: 7 5 X
Memory: 7 5 9
Memory: 4 5 9
Memory: 4 3 9
Memory: 4 3 7
Memory: 9 3 7
Memory: 9 6 7
Memory: 9 6 2
Memory: 1 6 2
Total Faults = 10
```

2. Least Recently Used (LRU) Replacement Algorithm

```c
#include <stdio.h>
int main()
{
    int n, i, j, cnt = 0, x, val;
    printf("enter the size of the pg table ");
    scanf("%d", &n);
    int a[n];
    int f = 0, l;
    printf("enter the size of demand paging list ");
    scanf("%d", &l);
    int ent[l];
    printf("enter the entries ");
    for (i = 0; i < l; i++){
        scanf("%d", &ent[i]);
    }
    for (j = 0; j < n; j++){
        a[i] = -1;
    }
    for (i = 0; i < l; i++){
        f = 0;
        for (j = 0; j < n; j++){
            if (a[j] == ent[i]){
                val = a[j];
                for (x = j; x < n - 1; x++)
                {
                    a[x] = a[x + 1];
                }
                a[x] = val;
                f = 1;
                break;
            }
        }
        if (f == 0){
            for (j = 0; j < n - 1; j++){
                a[j] = a[j + 1];
            }
        }
        else
            a[j] = ent[i];
        cnt++;
        continue;
    }
    printf("page fault is %d ", cnt);
}
```

OUTPUT-

```
enter the size of the pg table 3
enter the size of demand paging list 10
enter the entries 7
5 9 4 3 7 9 6 2 1
page fault is 10
```