

OPTIMAL PAGE REPLACEMENT ALGORITHM

```
import java.util.Scanner;
import java.io.IOException;

public class OptimalPageReplacement{
    public static void main(String[] args) throws IOException{
        Scanner in = new Scanner(System.in);

        int frames = 0;
        int pointer = 0;
        int numFault = 0;
        int ref_len;
        boolean isFull = false;
        int buffer[];
        boolean hit[];
        int fault[];
        int reference[];
        int mem_layout[][];

        System.out.println("Vishakha Sable");
        System.out.println("Roll No.: TECA151");
        System.out.println("Please enter the number of frames: ");
        frames = Integer.parseInt(in.nextLine());
        System.out.println("Please enter the length of the reference string: ");
        ref_len = Integer.parseInt(in.nextLine());
        reference = new int[ref_len];
        mem_layout = new int[ref_len][frames];
        buffer = new int[frames];
        hit = new boolean[ref_len];
        fault = new int[ref_len];
        for(int j = 0; j < frames; j++){
```

```

buffer[j] = -1;}

System.out.println("Please enter the reference string (hit Enter/Return after each number in the string):
");

for(int i = 0; i < ref_len; i++) {

reference[i] = Integer.parseInt(in.nextLine());}

System.out.println();

for(int i = 0; i < ref_len; i++){

int search = -1;

for(int j = 0; j < frames; j++){

if(buffer[j] == reference[i]){

search = j;

hit[i] = true;

fault[i] = numFault;

break;}}

if(search == -1){

if(isFull){

int index[] = new int[frames];

boolean index_flag[] = new boolean[frames];

for(int j = i + 1; j < ref_len; j++){

for(int k = 0; k < frames; k++){

if((reference[j] == buffer[k]) && (index_flag[k] == false)){

index[k] = j;

index_flag[k] = true;

break;}}}

int max = index[0];

pointer = 0;

if(max == 0) {

max = 200;}

for(int j = 0; j < frames; j++){

```

```

if(index[j] == 0){
index[j] = 200;}

if(index[j] > max){
max = index[j];
pointer = j;}}

buffer[pointer] = reference[i];

numFault++;

fault[i] = numFault;

if(!isFull){
pointer++;

if(pointer == frames){
pointer = 0;
isFull = true;} }}

for(int j = 0; j < frames; j++){
mem_layout[i][j] = buffer[j];}}

for(int i = 0; i < ref_len; i++){
System.out.print(reference[i] + ": Memory is: ");

for(int j = 0; j < frames; j++){

if (mem_layout[i][j] == -1){
System.out.printf("%3s ", "*");}

else{
System.out.printf("%3d ", mem_layout[i][j]);}}

System.out.print(": ");

if (hit[i]) {
System.out.print("Hit");}

else{
System.out.print("Page Fault");}

System.out.print(": (Number of Page Faults: " + fault[i] + ")");

System.out.println();}

```

```
System.out.println("Total Number of Page Faults: " + numFault);}}
```

OUTPUT:

```
Output Clear
java -cp /tmp/EnieEEXvb/OptimalPageReplacement
Vishakha Sable
Roll No.: TECA151
Please enter the number of frames:
3
Please enter the length of the reference string:
2
Please enter the reference string (hit Enter/Return after each number in the
string):
1
2

1: Memory is:  1  *  * : Page Fault: (Number of Page Faults: 1)
2: Memory is:  1  2  * : Page Fault: (Number of Page Faults: 2)
Total Number of Page Faults: 2

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