Optimal Page Replacement Algorithm

import java.util.Scanner;

import java.io.IOException;

public class Optimal

{

    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("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);

        System.out.println("Page Faults ratio : " + numFault+":"+ref\_len);

        in.close();

    }

}

OUTPUT :

Please enter the number of frames:

3

Please enter the length of the reference string:

12

Please enter the reference string (hit Enter/Return after each number in the string):

2

3

2

1

5

2

4

5

3

2

5

2

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

3: Memory is: 2 3 \* : Page Fault: (Number of Page Faults: 2)

2: Memory is: 2 3 \* : Hit: (Number of Page Faults: 2)

1: Memory is: 2 3 1 : Page Fault: (Number of Page Faults: 3)

5: Memory is: 2 3 5 : Page Fault: (Number of Page Faults: 4)

2: Memory is: 2 3 5 : Hit: (Number of Page Faults: 4)

4: Memory is: 4 3 5 : Page Fault: (Number of Page Faults: 5)

5: Memory is: 4 3 5 : Hit: (Number of Page Faults: 5)

3: Memory is: 4 3 5 : Hit: (Number of Page Faults: 5)

2: Memory is: 2 3 5 : Page Fault: (Number of Page Faults: 6)

5: Memory is: 2 3 5 : Hit: (Number of Page Faults: 6)

2: Memory is: 2 3 5 : Hit: (Number of Page Faults: 6)

Total Number of Page Faults: 6

Page Faults ratio : 6:12