

Assignment Title: Implementation of concept called Paging, using simulation.

Problem Statement: Write a Java Program (using OOP features) to implement paging simulation using

1. Least Recently Used (LRU)
2. Optimal algorithm.

CODE:

FIFO

```
import java.io.*;
public class FIFO {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int frames, pointer = 0, hit = 0, fault = 0, ref_len;
        int buffer[];
        int reference[];
        int mem_layout[][];

        System.out.println("Enter the Number of Frames: ");
        frames = Integer.parseInt(br.readLine());

        System.out.println("Enter the Length of Reference String: ");
        ref_len = Integer.parseInt(br.readLine());

        reference = new int[ref_len];
        mem_layout = new int[ref_len][frames];
        buffer = new int[frames];

        for (int j = 0; j < frames; j++)
            buffer[j] = -1;

        System.out.println("Please enter the reference string: ");
        for (int i = 0; i < ref_len; i++) {
            reference[i] = Integer.parseInt(br.readLine());
            int search = -1;

            for (int j = 0; j < frames; j++) {
                if (buffer[j] == reference[i]) {
                    search = j;
                    hit++;
                    break;
                }
            }

            if (search == -1) {
                buffer[pointer] = reference[i];
                fault++;
                pointer++;
                if (pointer == frames)
                    pointer = 0;
            }
        }
    }
}
```

```

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

System.out.println("\nMemory Layout:");
for (int i = 0; i < frames; i++) {
    for (int j = 0; j < ref_len; j++) {
        System.out.printf("%3d ", mem_layout[j][i]);
    }
    System.out.println();
}

System.out.println("\nThe number of Hits: " + hit);
System.out.println("Hit Ratio: " + (float) hit / ref_len);
System.out.println("The number of Faults: " + fault);
}
}

```

LRU

```

import java.io.*;

public class LRU {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        int frames, hit = 0, fault = 0, ref_len;
        int buffer[];
        int reference[];
        int mem_layout[][];
        int counter[];

        System.out.print("Enter the Number of Frames: ");
        frames = Integer.parseInt(br.readLine());

        System.out.print("Enter the Length of Reference String: ");
        ref_len = Integer.parseInt(br.readLine());

        reference = new int[ref_len];
        mem_layout = new int[ref_len][frames];
        buffer = new int[frames];
        counter = new int[frames];

        for (int j = 0; j < frames; j++) {
            buffer[j] = -1;
            counter[j] = 0;
        }

        System.out.println("Please enter the reference string: ");
        for (int i = 0; i < ref_len; i++) {
            reference[i] = Integer.parseInt(br.readLine());
            int search = -1;
            for (int j = 0; j < frames; j++) {
                if (buffer[j] == reference[i]) {

```

```

        search = j;
        hit++;
        counter[j] = i + 1;
        break;
    }
}
if (search == -1) {
    int min = 0;
    for (int j = 1; j < frames; j++) {
        if (counter[j] < counter[min]) {
            min = j;
        }
    }
    buffer[min] = reference[i];
    counter[min] = i + 1;
    fault++;
}
for (int j = 0; j < frames; j++) {
    mem_layout[i][j] = buffer[j];
}
System.out.println("\nMemory Layout:");
for (int i = 0; i < frames; i++) {
    for (int j = 0; j < ref_len; j++) {
        System.out.printf("%3d ", mem_layout[j][i]);
    }
    System.out.println();
}
System.out.println("\nThe number of Hits: " + hit);
System.out.println("Hit Ratio: " + (float) hit / ref_len);
System.out.println("The number of Faults: " + fault);
}
}

```

OPTIMAL

```

import java.io.*;

public class Optimal {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        int frames, hit = 0, fault = 0, ref_len;
        int buffer[];
        int reference[];
        int mem_layout[][];

        System.out.print("Enter the Number of Frames: ");
        frames = Integer.parseInt(br.readLine());

        System.out.print("Enter the Length of Reference String: ");
        ref_len = Integer.parseInt(br.readLine());

        reference = new int[ref_len];
    }
}

```

```

mem_layout = new int[ref_len][frames];
buffer = new int[frames];

for (int j = 0; j < frames; j++)
    buffer[j] = -1;

System.out.println("Please enter the reference string: ");
for (int i = 0; i < ref_len; i++) {
    reference[i] = Integer.parseInt(br.readLine());
    int search = -1;
    for (int j = 0; j < frames; j++) {
        if (buffer[j] == reference[i]) {
            search = j;
            hit++;
            break;
        }
    }

    if (search == -1) {
        int pos = -1, farthest = i;
        for (int j = 0; j < frames; j++) {
            int next_use = -1;
            for (int k = i + 1; k < ref_len; k++) {
                if (buffer[j] == reference[k]) {
                    next_use = k;
                    break;
                }
            }
            if (next_use == -1) {
                pos = j;
                break;
            }
            if (next_use > farthest) {
                farthest = next_use;
                pos = j;
            }
        }
        buffer[pos] = reference[i];
        fault++;
    }
}

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

System.out.println("\nMemory Layout:");
for (int i = 0; i < frames; i++) {
    for (int j = 0; j < ref_len; j++) {
        System.out.printf("%3d ", mem_layout[j][i]);
    }
    System.out.println();
}

System.out.println("\nThe number of Hits: " + hit);

```

```

        System.out.println("Hit Ratio: " + (float) hit / ref_len);
        System.out.println("The number of Faults: " + fault);
    }
}

```

OUTPUT:

FIFO

```
CODE\Page-Replacement-Algotithm\FIFO on \ main [!?] via v24.0.2
> javac FIFO.java
```

```
CODE\Page-Replacement-Algotithm\FIFO on \ main [!?] via v24.0.2
> java FIFO.java
Enter the Number of Frames:
3
Enter the Length of Reference String:
12
Please enter the reference string:
7
0
1
2
0
3
0
4
2
3
0
3

Memory Layout:
 7 7 7 2 2 2 2 4 4 4 0 0
 -1 0 0 0 0 3 3 3 2 2 2 2
 -1 -1 1 1 1 0 0 0 3 3 3 3
```

```
The number of Hits: 2
Hit Ratio: 0.16666667
The number of Faults: 10
```

LRU

```
CODE\Page-Replacement-Algotithm\LRU on \ main [!?] via v24.0.2
> javac LRU.java
```

```
CODE\Page-Replacement-Algotithm\LRU on \ main [!?] via v24.0.2
> java LRU.java
Enter the Number of Frames: 3
Enter the Length of Reference String: 12
Please enter the reference string:
7
0
1
2
0
3
0
4
2
3
0
3

Memory Layout:
 7 7 7 2 2 2 2 4 4 4 0 0
 -1 0 0 0 0 0 0 0 3 3 3 3
 -1 -1 1 1 1 3 3 3 2 2 2 2
```

```
The number of Hits: 3
Hit Ratio: 0.25
The number of Faults: 9
```

OPTIMAL

```
CODE\Page-Replacement-Algotithm\Optimal on ▶ main [!?] via v24.0.2
› javac Optimal.java
```

```
CODE\Page-Replacement-Algotithm\Optimal on ▶ main [!?] via v24.0.2
› java Optimal.java
```

Enter the Number of Frames: 3

Enter the Length of Reference String: 12

Please enter the reference string:

```
7
0
1
2
0
3
0
4
2
3
0
3
```

Memory Layout:

7	0	0	0	0	0	0	0	0	0	0	0	0
-1	-1	1	2	2	3	3	4	2	3	3	3	3
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

The number of Hits: 4

Hit Ratio: 0.33333334

The number of Faults: 8