

## 1)FIFO

```
import java.util.Scanner;

class fifo
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);

        int len;

        System.out.println("Enter length of reference string: ");

        len = sc.nextInt();

        int[] incomingStream = new int[len];

        System.out.println("Enter reference string :");

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

            incomingStream[i] = sc.nextInt();

        }

        int pageFaults = 0;

        System.out.println("Enter no. of frames: ");

        int frames = sc.nextInt();

        int m, n, s, pages;

        pages = len;

        System.out.println("Incoming\tFrame 1 \tFrame 2 \tFrame 3");

        int[] temp = new int[frames];

        for(m = 0; m < frames; m++)

        {

            temp[m] = -1;

        }

    }
}
```

```

for(m = 0; m < pages; m++){
    s = 0;
    for(n = 0; n < frames; n++) {
        if(incomingStream[m] == temp[n]) {
            s++;
            pageFaults--;
        }
    }
    pageFaults++;
    if((pageFaults < frames) && (s == 0)) {
        temp[m] = incomingStream[m];
    }
    else if(s == 0) {
        temp[(pageFaults - 1) % frames] = incomingStream[m];
    }

    System.out.println();
    System.out.print(incomingStream[m] + "\t\t");
    for(n = 0; n < frames; n++) {
        if(temp[n] != -1)
            System.out.print(temp[n] + "\t\t");
        else
            System.out.print(" - \t\t");
    }
}

System.out.println("\nTotal Page Faults:\t" + pageFaults);
System.out.println("\nTotal Page Hits:\t" + (len-pageFaults));
}

```

```
}
```

## 2)LRU

```
import java.io.*;
import java.util.*;
public class LRU {
    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;
        boolean isFull = false;
        int[] buffer;
        ArrayList<Integer> stack = new ArrayList<Integer>();
        int[] reference;
        int[][] mem_layout;
        System.out.println("Please enter the number of Frames: ");
        frames = Integer.parseInt(br.readLine());
        System.out.println("Please enter the length of the 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());
        }
    }
}
```

```

System.out.println();
for (int i = 0; i < ref_len; i++) {
    if (stack.contains(reference[i])) {
        stack.remove(stack.indexOf(reference[i]));
    }
    stack.add(reference[i]);
    int search = -1;
    for (int j = 0; j < frames; j++) {
        if (buffer[j] == reference[i]) {
            search = j;
            hit++;
            break;
        }
    }
    if (search == -1) {
        if (isFull) {
            int min_loc = ref_len;
            for (int j = 0; j < frames; j++) {
                if (stack.contains(buffer[j])) {
                    int temp = stack.indexOf(buffer[j]);
                    if (temp < min_loc) {
                        min_loc = temp;
                        pointer = j;
                    }
                }
            }
        }
        buffer[pointer] = reference[i];
        fault++;
        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 < frames; i++) {
    for (int j = 0; j < ref_len; j++)
        System.out.printf("%3d ", mem_layout[j][i]);
    System.out.println();
}

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

```

### 3)OPTIMAL

```

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: ");

```

```

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] + "---->");
    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.println();  
}  
System.out.println("Total Number of Page Faults: " + numFault);  
System.out.println("Total Number of Page hits: " + (ref_len-numFault));  
}  
}
```