

```
package ass7optimalpagereplacement;
```

```
import java.util.Scanner;
```

```
public class ass7optimalpagereplacement {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int noofpages, capacity, ptr = 0, hit = 0, fault = 0;
        boolean isFull = false;
        double hitRatio, faultRatio;
        System.out.print("Enter the number of pages you want to enter: ");
        noofpages = sc.nextInt();
        int pages[] = new int[noofpages];
        for (int i = 0; i < noofpages; i++) {
            pages[i] = sc.nextInt();
        }
        System.out.print("Enter the capacity of frame: ");
        capacity = sc.nextInt();
        int frame[] = new int[capacity];
        int table[][] = new int[noofpages][capacity];
        for (int i = 0; i < capacity; i++) {
            frame[i] = -1;
        }
        System.out.println("-----");
        for (int i = 0; i < noofpages; i++) {
            int search = -1;
            for (int j = 0; j < capacity; j++) {
                if (frame[j] == pages[i]) {
                    search = j;
                    hit++;
                    System.out.printf("%4s", "H");
                    break;
                }
            }
            if (search == -1) {
                if (isFull) {
                    int[] index = new int[capacity];
                    boolean[] index_flag = new boolean[capacity];
                    for (int j = i + 1; j < noofpages; j++) {
                        for (int k = 0; k < capacity; k++) {
                            if ((pages[j] == frame[k]) &&
                                (!index_flag[k])) {
                                index[k] = j;
                                index_flag[k] = true;
                                break;
                            }
                        }
                    }
                }
                int max = index[0];
                ptr = 0;
                if (max == 0)
                    max = 200;
                for (int j = 0; j < capacity; j++) {
                    if (index[j] == 0)
                        index[j] = 200;
                }
            }
        }
    }
}
```

```

        if (index[j] > max) {
            max = index[j];
            ptr = j;
        }
    }
}
frame[ptr] = pages[i];
fault++;
System.out.printf("%4s", "F");
if (!isFull) {
    ptr++;
    if (ptr == capacity) {
        ptr = 0;
        isFull = true;
    }
}
}
System.arraycopy(frame, 0, table[i], 0, capacity);
}
System.out.println("\n-----");
for (int i = 0; i < capacity; i++) {
    for (int j = 0; j < noofpages; j++)
        System.out.printf("%3d ", table[j][i]);
    System.out.println();
}
System.out.println("-----");
hitRatio = ((double) hit / noofpages) * 100;
faultRatio = ((double) fault / noofpages) * 100;
System.out.println("Page Fault: " + fault + "\nPage Hit: " + hit);
System.out.printf("Hit Ratio: %.2f \nFault Ratio: %.2f ", hitRatio, faultRatio);
sc.close();
}
}

```