Program 2: Page Replacement Algorithm

(Both hardcopy and electronic copy are due 5:00 pm on Friday in 9th week, turnin hardcopy in Dayi’s mail box. No late turn-in is accepted.)

Note: This is a group assignment with each group having 1-2 members. Each group should only turn in one hard copy and one electronic copy. Every group member will be given the same grade.

This project is composed of two parts: writing a simulator in Java comparing four different page replacement algorithms, optimal algorithm, FIFO, LRU and second chance, and analyzing the data collected in the simulation. For extra credits, you also need to implement ARB algorithm and compare it with the others (details about extra credits will be described in the section of “what to turn in”).

Page reference string:
There are 8 user jobs. For each job, the input to your program will be the page reference strings from that job. The file /cs/classes/www/05S/cis415/cis415/turnin/program2/paging.input[1-8] contains the reference strings for the user programs in the following format.

Line 1: # JOB ID #
Line 2: total number of distinct pages in the job
Line 3: total number of page references in the entire reference string
Lines 4-n: reference string in decimal separated by blank or newline.
Last line: # END of JOB ID #

Output:
For each page replacement algorithm, your program will record the page fault rate for frame allocations varying from 10% to 90% of the total number of distinct pages required by the job in increments of 10%. The page fault rate is defined as the number of page faults divided by the total number of references in the page reference string. Please count every page faults from the loading of the first page of the program into memory. This is repeated for each of the 8 user jobs.

The output of your program consists of one table for each page replacement algorithm that shows the page fault rate for each combination of user job and frame allocation. In addition, you will print out a summary table giving the average page fault rate over all the user jobs.

A sample output is as following, which is only used to show you the format of the output.
What to turn in

a. Java classes:

[doubleLink.java]: The double link data structure will be used by the LRU algorithm. FIFO can also use this data structure.

Inside doubleLink.java, a node class is defined as following:

```java
public class linkNode {
    public:
    int item;
    linkNode left;
    linkNode right;
}
```

You can add any method, which you think is necessary to your implementation.

[circularLink.java]: The circular link data structured is used by the second chance algorithm. circularLink uses the same node class defined above. You define methods which are necessary.

[Algorithm.java]: There are at least four methods in “Algorithm” class: OPT(reference string), FIFO(reference string), secondChance (reference string) and LRU (reference string). Each represents one page replacement algorithm and returns the number of page faults using that algorithm. For extra credits, another method ARB is required to be implemented. Refer to the textbook for the details of the algorithms.

[Simulator.java]: The driver class, which contains page table simulated by an array or a vector. Different methods in Algorithm will be called from here.

b. Documents:

1. Your well-documented source code, and hardcopy of the output produced by your program consisting of one table for each page replacement algorithm.
2. Draw a graph to compare performance of different algorithms (FIFO, second chance, LRU and optimal algorithm) with different number of frames. The x-axis is frame
allocation rate varying from 10% to 90%, and the y-axis is the page fault rate. Page fault rate = # of page faults / total # of page references.

3. Analyze how the algorithms perform compared with each other and the reasons for that.

4. Extra Credit: Analyze Additional-Reference-Bits algorithm by varying the number of history bits (3,5,9 bits). Let the updating interval be every 3 memory references. When there are multiple pages with the same lowest number, use FIFO selection among them. Graph the results and compare with LRU and Second Chance only. Draw three graphs to compares ARB-3, ARB-5, ARB-9 with LRU and Second Chance separately.

Instructions for electronic turn in:

1. Create a tar file of your program's files. Assuming you are in the directory where your files are stored, you can run the following command:
   ```bash
gtar -czvf pa2.tar.gz *
   ```

2. Submit the compressed tar file:

   ```bash
   /cs/classes/www/05S/cis415/cis415/turnin/turnin -p 8888 -s ix.cs.uoregon.edu pa2 pa2.tar.gz
   ```

You should be able to go to http://www.cs.uoregon.edu/classes/05S/cis415/cis415/turnin/ and check the names, sizes, and dates of the files you have submitted, but NOT their contents.