# Project Part 1 exam: Memory Management: Victim Replacement Algorithms

## Instructions

* **MAKE SURE TO USE THE EXAM VERSION OF THE SIMULATOR SimVictimExam.py**
* **EDIT THIS document to insert your answers**
* **Time: 1 hour - Submit your completed document by the end of the hour on the submission link provided. DO NOT submit a PDF.**
* **Where necessary copy and paste images of the data in question throughout your document to illustrate your answers.**
* **If necessary, re-run the program but make sure to show the new data in your answer.**

# Paste the ID and date and timestamp from the simulator here:

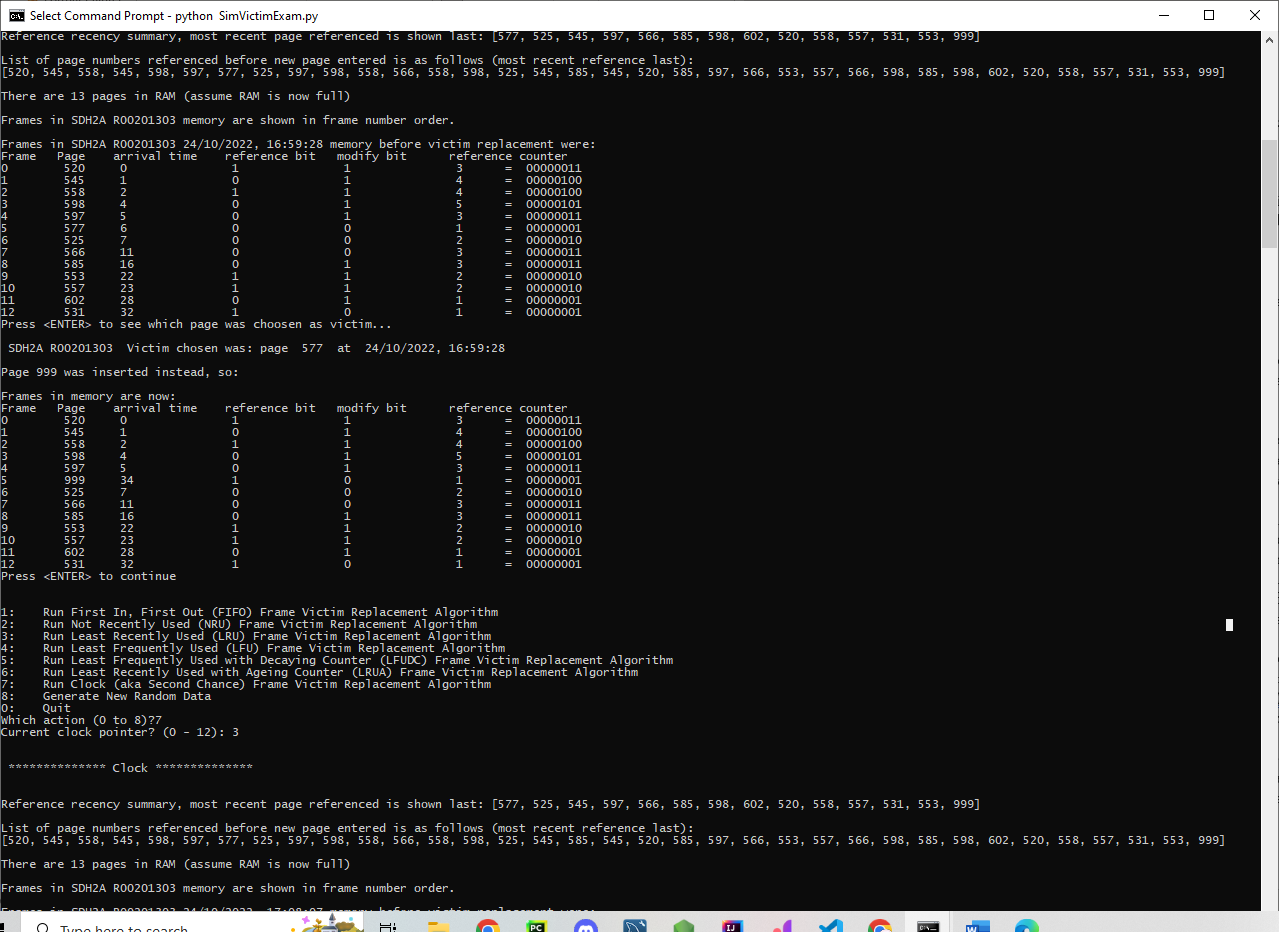
Text

Description automatically generated

## Answer both questions

Run each algorithm listed below in turn. For each one, copy and paste **ALL** of the output and give the explanations indicated. **In all cases use examples from the data generated to illustrate you answers. Explanations that do not refer to the data are not acceptable.**

1. LFU (30 marks)  
**Fully explain** why the victim chosen was picked with reference to the specific data used.

LFU: Least Frequently Use 

ANS: 999 was the page victim that is the least frequently used. LFU Count the approximate number of references by periodically adding the reference bit to a reference counter byte and clearing the reference bit also the page with the lowest reference counter.

Advantages –

Easy to implement = The counter records the number of references in this case it was 00000001

Disadvantages –

Only takes the frequency of reference into account and ignores recency of reference. A page that was heavily used in the past and is now useless i.e an old glory page, can avoid replacement.

LFU with Decaying Count:

As for LFU, but reference count is periodically divided by 2. Page 999 is even with the lowest decayed reference counter, LFU is easily implement here.again the most biggest advantage for this is also that its:

Easy to implement. Deals with the ‘old glory’ page as its count will decay. But recent arrivals have not had a chance to rack up many references.

The decayed counter records the number of references:

00000000 this number indicates there are no references or a page has just arrived in the memory.in page 999: 00000001 indicates few references but has it few because it has decayed or is it a new page. Reference history ‘decaying count’ the byte is manipulated every time period (every 5 page references): it is incremented if the page is referenced in the last time period but is divided by two if it is not. This way only pages that were referenced often and recently will have high scores. Page 999 is among those pages that has lowest reference number like page 602 and 531 which indicates that this page is also LRU(least recently used).

2. Clock (70 marks)

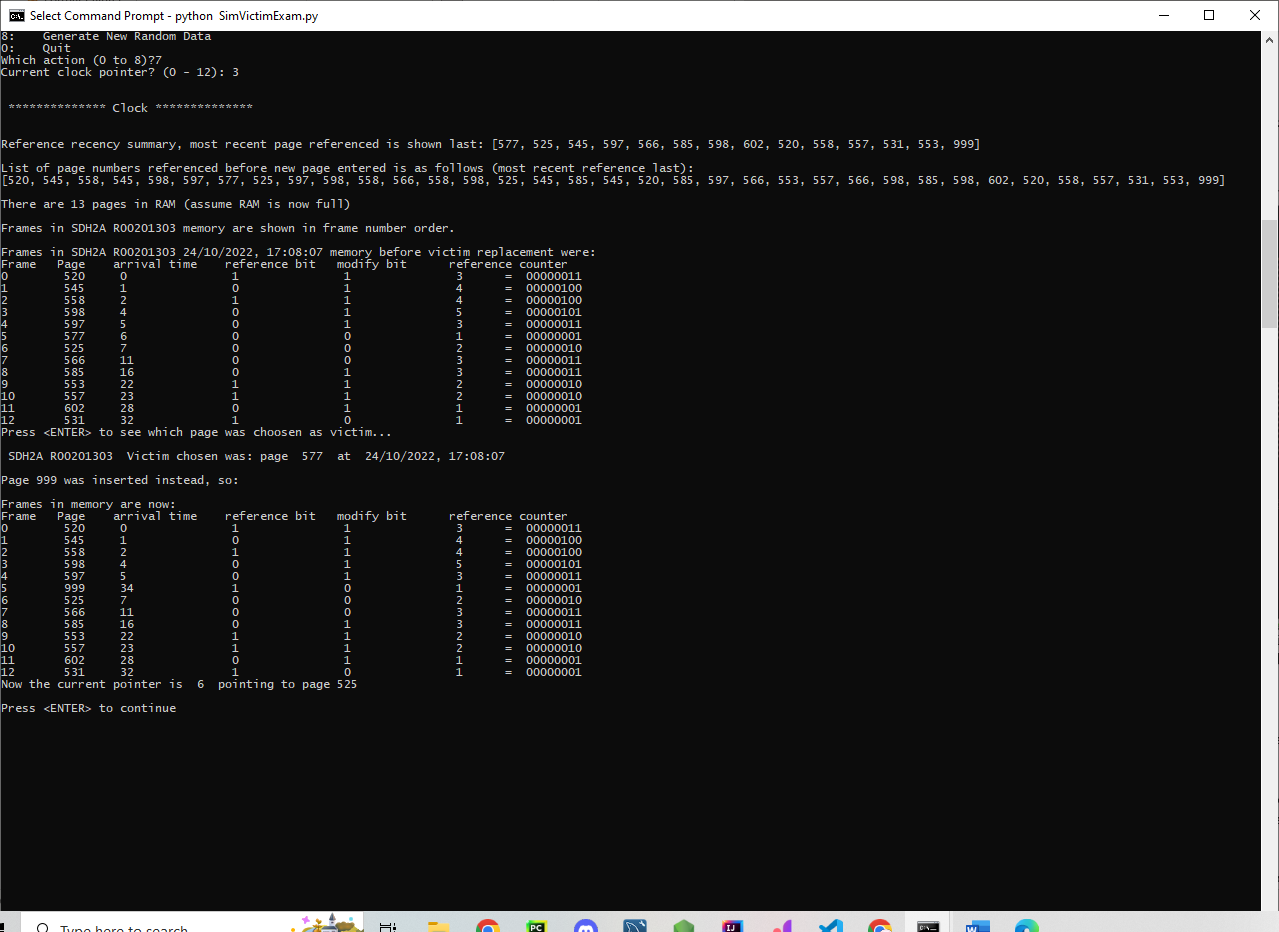
i. Fully explain why the victim chosen was picked with reference to the specific data used

ii. Fully explain the meaning of the clock pointer value

iii. Fully explain why the pointer value is now what is indicated in your output

The 3 operator turns a value of type pointer to T into a variable of type T

iv. Fully explain any changes in the reference bit values of other pages in your output once the new page has been installed in RAM

Clock /Second Chance 

Ans: 999 is a page that has not been referenced recently and is the doesn’t need to be written to disk. Maintains a circular list of page frames with a pointer to the next possible victim. A page stored in a frame will only be considered for replacement when the pointer comes around to it. Once considered that frame’s page will not be considered again until all other frames have been considered. This algorithm keeps of information about each page:

• reference bit = A bit that is indicating if the page was referenced

since the last periodic reset of reference bits to 0. If = 1 it means the page

has been referenced recently. as for this 999 page dirty bit value = 1.

• dirty bit = A dirty bit or modified bit is a bit that is associated with a block of computer memory and indicates whether the corresponding block of memory has been modified.as for this 999 page dirty bit value = 0.

Clock algorithm uses a circular data structure to record this information about the pages in the frames. Every time a page must be replaced, repeat the following until victim chosen: to examine the page in the next candidate frame (i.e. the page stored in the frame at the pointer):

if its reference bit is clear

then

if its ‘dirty bit’ is clear

then

replace this page’s ID and information with the incoming page’s equivalent in the circular list;

move the pointer to the next page frame in the list;

replace the victim in the frame in RAM with the incoming page.

Else

move the pointer to the next page frame in the list and repeat;

endif

else

clear the page’s reference bit (second chance)

move the pointer to the next page frame in the list and repeat;

endif

Takes recency and convenience into account but also examines frames in turns:

• Turns – pages examined in ‘turn’ order

• Recency – pages whose reference bit is set are given second chance

• Convenience - avoids copying modified pages.

This is a variation on the FIFO algorithm.

If a page gets a second chance its reference bit is cleared and the victim pointer moves (clockwise) on to the next candidate on the circular list. Therefore a page which has been given a second chance ex.999 will not be replaced until all other pages have been replaced or given their second chance and then only if it does not get another second chance. Thus, if a page is used often enough to keep its reference bit set, then it will never be replaced.Create an array frames to track the pages currently in memory and another Boolean array second\_chance to track whether that page has been accessed since it’s last replacement (that is if it deserves a second chance or not) and a variable pointer to track the target for replacement.