## Execution

- Navigate to the root directory of the folder and execute the following commands in the terminal

`g++ fifo.cpp -o fifo`
`g++ lru.cpp -o lru`
`g++ opt.cpp -o opt`
`./fifo`
`./lru`
`./opt`

- Each of the 3 programs prompt for 3 inputs
  - Number of simulations
  - Number of pages in our reference string
  - Upper limit of frame size

Description

- Approach taken is different as program instead of taking reference string from the user, generates the reference string itself randomly. Moreover instead of taking the frame size, it takes the upper limit of frame size as input; it then processes the reference string with value of frame size varying between 3 and the upper limit.
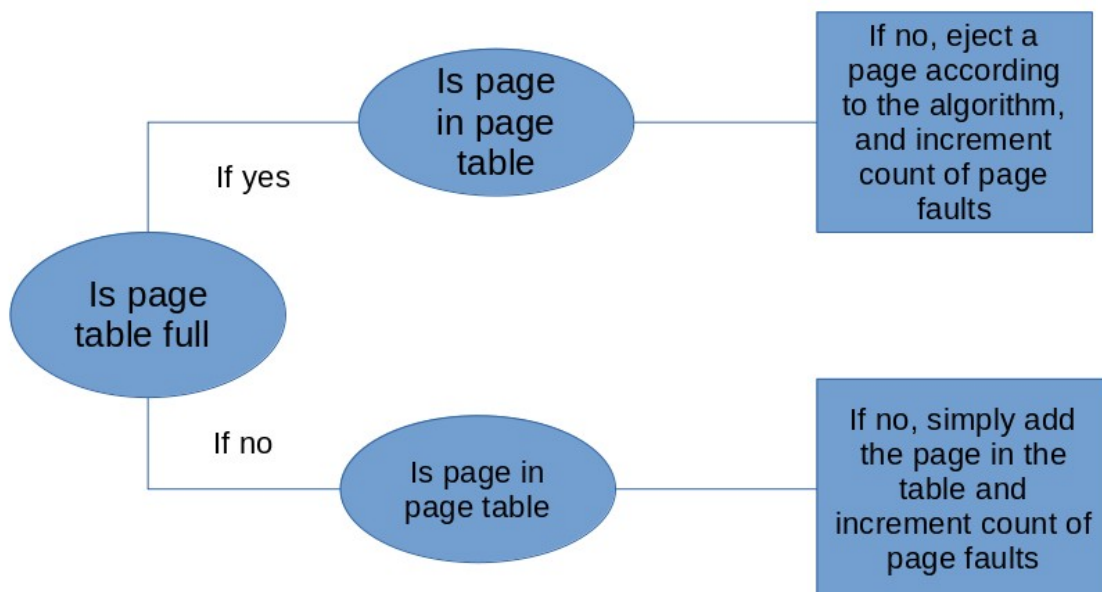
- Member methods
  - **generate_ref_string(int length):** here length is the number of pages in our reference string. The function generates reference string where page numbers can vary between 0 and 30.
    - Instead of generating a simple number string, it generates **'.'** separated number string: x.x.x.x.x.x.x.x.x.x.x where x can be any number between 1 and 30.
    - This is better as naive generation can have only 10 unique pages instead of 30; therefore for frame sizes greater than 9, page faults will always going to be 10.

- Infact the number 30 was taken only for demonstration purposes. It can be any number; making our analysis modular to change and extension
  - **count_lru_page_faults(** reference_string, frame_size**)**
  - **count_opt_page_faults(** reference_string, frame_size**)**
    - FIFO implementation was done in main()

Analysis

With 100 pages long reference strings, max frame size of 15 and running the simulation for 100 times, following results were reported: clearly optimal page replacement algoritm works best. For each simulation; reference string is re-initialized.

```
ceyxasm@pop-os:~/.../CSL3030/lab_6$ ./fifo
Enter the pages to be in our reference string: 100
enter the upper limit on frame size ULFS (for a parti
cular reference string; frame size will be varied fro
m 3 to UFSL): 15
Enter the number of times execution should be run: 10
0
Average page faults for frame size 3 is 91
Average page faults for frame size 4 is 86
Average page faults for frame size 5 is 83
Average page faults for frame size 6 is 78
Average page faults for frame size 7 is 72
Average page faults for frame size 8 is 69
Average page faults for frame size 9 is 64
Average page faults for frame size 10 is 59
Average page faults for frame size 11 is 57
Average page faults for frame size 12 is 52
Average page faults for frame size 13 is 51
Average page faults for frame size 14 is 51
Average page faults for frame size 15 is 49
ceyxasm@pop-os:~/.../CSL3030/lab_6$
```

```
ceyxasm@pop-os:~/.../CSL3030/lab_6$ ./lru
Enter the number of simulations: 100
Enter the pages to be in our reference string: 100
enter the upper limit on frame size ULFS (for a particu
lar reference string; frame size will be varied from 3
to UFSL): 15
average page faults for frame size 3: 91
average page faults for frame size 4: 88
average page faults for frame size 5: 85
average page faults for frame size 6: 81
average page faults for frame size 7: 79
average page faults for frame size 8: 76
average page faults for frame size 9: 73
average page faults for frame size 10: 70
average page faults for frame size 11: 67
average page faults for frame size 12: 64
average page faults for frame size 13: 61
average page faults for frame size 14: 58
average page faults for frame size 15: 55
ceyxasm@pop-os:~/.../CSL3030/lab_6$
```

```
ceyxasm@pop-os:~/.../CSL3030/lab_6$ ./opt
Enter number of simulations: 100
Enter the upper limit of frame size( for a particular
 ref str; frame size varies from 3 to this upper limi
t): 15
Enter number of pages: 100
Average page faults for frame size 3: 74
Average page faults for frame size 4: 68
Average page faults for frame size 5: 62
Average page faults for frame size 6: 58
Average page faults for frame size 7: 54
Average page faults for frame size 8: 50
Average page faults for frame size 9: 48
Average page faults for frame size 10: 45
Average page faults for frame size 11: 43
Average page faults for frame size 12: 41
Average page faults for frame size 13: 39
Average page faults for frame size 14: 37
Average page faults for frame size 15: 36
ceyxasm@pop-os:~/.../CSL3030/lab_6$
```

Belady's Anomaly

The phenomenon in which increasing the page frame results in an increase in the number of page faults for a given memory access pattern.

Belady's anomaly out of the three can only be seen in FIFO as the other two belong to class of stack based algorithm and can never suffer from Belady. Moreover, it is not that every reference string results in the anomaly for FIFO; but certain reference string do worsen FIFO performance on increasing the number of frames.

After a couple of hit and trials, Belady anomaly was shown by FIFO as shown below.
Here increasing the page frames from 14 to 15 saw an increase in number of page faults from 47 to 49. Unexpectedly, re-running the simulation did not result in the anomaly.

```
ceyxasm@pop-os:~/.../CSL3030/lab_6$ ./fifo
Enter the pages to be in our reference string: 100
enter the upper limit on frame size ULFS (for a particular reference string; frame size will be varied from 3 to UFSL): 15
Enter the number of times execution should be run: 1
Average page faults for frame size 3: 91
Average page faults for frame size 4: 86
Average page faults for frame size 5: 83
Average page faults for frame size 6: 78
Average page faults for frame size 7: 72
Average page faults for frame size 8: 69
Average page faults for frame size 9: 64
Average page faults for frame size 10: 57
Average page faults for frame size 11: 59
Average page faults for frame size 12: 52
Average page faults for frame size 13: 51
Average page faults for frame size 14: 47
Average page faults for frame size 15: 49
Average page faults for frame size 16: 45
Average page faults for frame size 17: 44
Average page faults for frame size 18: 41
Average page faults for frame size 19: 39
Average page faults for frame size 20: 36
ceyxasm@pop-os:~/.../CSL3030/lab_6$
```