**Aim:** To implement FIFO Page Replacement Algorithm and calculate Hit ratio and miss

ratio.

**Algorithm:**

**Step 1:** Start to traverse the pages.

**Step 2:** If the memory holds fewer pages then the capacity go to Step 3, else go to

Step 5.

**Step 3:** Push pages in the queue one at a time until the queue reaches its maximum

capacity or all page requests are fulfilled.

**Step 4:** If the current page is present in the memory, do nothing.

**Step 5:** Else, pop the topmost page from the queue as it was inserted first.

**Step 6:** Replace the topmost page with the current page from the string.

**Step 7:** Increment the page faults.

**Code:**

#include <stdio.h>

int main()

{

int referenceString[10], pageFaults = 0, m, n, s, pages, frames;

printf("\nEnter the number of Pages: ");

scanf("%d", &pages);

float pagesss=pages;

printf("\nEnter reference string values:\n");

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

{

printf("Value No. [%d]:\t", m + 1);

scanf("%d", &referenceString[m]);

}

printf("\n What are the total number of frames:\t");

{

scanf("%d", &frames);

}

int temp[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(referenceString[m] == temp[n])

{

s++;

pageFaults--;

}

}

pageFaults++;

if((pageFaults <= frames) && (s == 0))

{

temp[m] = referenceString[m];

}

else if(s == 0)

{

temp[(pageFaults - 1) % frames] = referenceString[m];

}

printf("\n");

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

{

printf("%d\t", temp[n]);

}

}

printf("\nTotal Page Faults: %d", pageFaults);

float faults=pageFaults;

float miss=faults/pagesss;

float hit=1-miss;

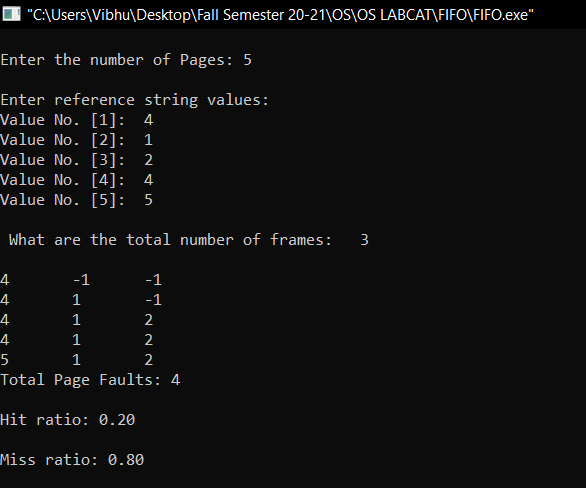
printf("\n\nHit ratio: %.2f",hit);

printf("\n\nMiss ratio: %.2f\n\n",miss);

return 0;

}

**Output(screenshots):**

****