**Aim:** To find the number of page faults, hit-ratio by implementing the Optimal Page Replacement Algorithm.

**Algorithm:**

**Step 1:**Push the first page in the stack as per the memory demand.

**Step 2:** Push the second page as per the memory demand.

**Step 3:** Push the third page until the memory is full.

**Step 4:** As the queue is full, the page which is least recently used is popped.

**Step 5:** Repeat step 4 until the page demand continues and until the processing is over.

**Step 6:** Terminate the program.

**Code:**

#include<stdio.h>

int main()

{

int no\_of\_pages, frames[10], pages[30], temp[10], flag1, flag2, flag3, i, j, k, pos, max, faults = 0,no\_of\_frames;

printf("Enter number of frames: ");

scanf("%d", &no\_of\_frames);

printf("Enter number of pages: ");

scanf("%d", &no\_of\_pages);

float pagesss=no\_of\_pages;

printf("Enter page reference string: ");

for(i = 0; i < no\_of\_pages; ++i){

scanf("%d", &pages[i]);

}

for(i = 0; i < no\_of\_frames; ++i){

frames[i] = -1;

}

for(i = 0; i < no\_of\_pages; ++i){

flag1 = flag2 = 0;

for(j = 0; j < no\_of\_frames; ++j){

if(frames[j] == pages[i]){

flag1 = flag2 = 1;

break;

}

}

if(flag1 == 0){

for(j = 0; j < no\_of\_frames; ++j){

if(frames[j] == -1){

faults++;

frames[j] = pages[i];

flag2 = 1;

break;

}

}

}

if(flag2 == 0){

flag3 =0;

for(j = 0; j < no\_of\_frames; ++j){

temp[j] = -1;

for(k = i + 1; k < no\_of\_pages; ++k){

if(frames[j] == pages[k]){

temp[j] = k;

break;

}

}

}

for(j = 0; j < no\_of\_frames; ++j){

if(temp[j] == -1){

pos = j;

flag3 = 1;

break;

}

}

if(flag3 ==0){

max = temp[0];

pos = 0;

for(j = 1; j < no\_of\_frames; ++j){

if(temp[j] > max){

max = temp[j];

pos = j;

}

}

}

frames[pos] = pages[i];

faults++;

}

printf("\n");

for(j = 0; j < no\_of\_frames; ++j){

printf("%d\t", frames[j]);

}

}

printf("\n\nTotal Page Faults = %d", faults);

float hit;

float miss=faults/pagesss;

hit=1-miss;

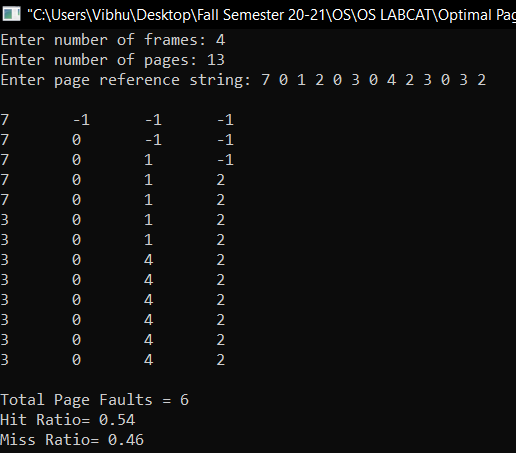
printf("\nHit Ratio= %.2f",hit);

printf("\nMiss Ratio= %.2f",miss);

return 0;

}

**Output(screenshots):**

****