# **33.Construct a C program to simulate the optimal paging technique of memory management**

**Aim:** To Construct a C program to simulate the optimal paging technique of memory management

# **Algorithm:**

* 1. Create an array to represent the page frames in memory.
  2. Initialize all page frames to -1, indicating that they are empty.
  3. Initialize a variable for page faults to zero.
  4. Read the reference string (sequence of page numbers) from the useror use a predefined array.
  5. For each page in the reference string, do the following:
  6. Check if the page is already in memory (a page hit).
  7. If it's a page hit, move to the next page.
  8. If it's a page fault (page not in memory), do the following:
  9. Increment the page fault count.
  10. Calculate the future references of each page in memory by scanningthe remaining part of the reference string.
  11. Find the page that will not be used for the longest time in the future(the optimal page to replace).
  12. Replace the optimal page with the new page.
  13. Continue this process for all pages in the reference string.
  14. After processing all pages, display the total number of page faults.

# **Program:**

#include<stdio.h> int main()

{

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

printf("Enter number of frames: ");

scanf("%d", &no\_of\_frames); printf("Enter number of pages: "); scanf("%d", &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);

return 0;

}

Output:

