PRACTICAL NO 6: PAGING [OPTIMAL METHOD]

```
#include <stdio.h>
#define INFINITE 1000
// Function to calculate the distance for OPT algorithm
int fdistance(int trace[], int ntrace, int start, int pageno) {
 int i;
 for (i = start + 1; i < ntrace; i++)
 if (pageno == trace[i])
 return (i - start);
 return INFINITE;
}
// Function to search for a page in the current frames
int search(int a[], int n, int pageno) {
 int i;
 for (i = 0; i < n; i++)
 if (a[i] == pageno)
 return 1;
 return 0;
}
// Function to find the frame that has the maximum distance
int findmax(int a[], int n) {
 int i, j;
 j = 0;
 for (i = 1; i < n; i++)
 if (a[i] > a[j])
 j = i;
 return j;
}
// Function to find an empty frame
int findempty(int a[], int n) {
 int i;
 for (i = 0; i < n; i++)
 if (a[i] == -1)
 return i;
 return -1;
}
// Main function to implement the OPT page replacement algorithm
int main() {
 int optf[10], trace[30], ntrace, nframes;
 int i, j, loc, optd[10];
 int page_faults = 0;
 printf("\nEnter number of frames: ");
 scanf("%d", &nframes);
 printf("\nEnter number of entries in the page trace: ");
 scanf("%d", &ntrace);
 printf("\nEnter page trace: ");
 for (i = 0; i < ntrace; i++)
 scanf("%d", &trace[i]);
 for (i = 0; i < nframes; i++) {
 optf[i] = -1; // Initialize frames to -1 (empty)
 optd[i] = 0; // Initialize distances to 0
 }
 printf("\nPage no. OPT Allocation\n");

 for (i = 0; i < ntrace; i++) {
 if (!search(optf, nframes, trace[i])) { // Page not found in frames
 loc = findempty(optf, nframes); // Look for an empty frame
 if (loc != -1) { // Empty frame found
 optf[loc] = trace[i];
 } else { // No empty frame, replace one
 loc = findmax(optd, nframes); // Find the frame with max
distance
 optf[loc] = trace[i]; // Replace it
```

```
 }
 page_faults++; // Increment page faults for the missed page
 }
 for (j = 0; j < nframes; j++) // Update distances for all
frames
 optd[j] = fdistance(trace, ntrace, i, optf[j]);
 // Print the current state of frames
 printf(" %d ", trace[i]);
 for (j = 0; j < nframes; j++)
 printf("%3d ", optf[j]);
 printf("\n");
 }
 printf("\nPAGE FAULTS: %d\n", page_faults);
 return 0;
}
```

OUTPUT:

```
Enter number of frames: 3

Enter number of entries in the page trace: 7

Enter page trace: 1
2
3
4
1
2
5

Page no.  OPT Allocation
 1        1  -1  -1
 2        1   2  -1
 3        1   2   3
 4        1   2   4
 1        1   2   4
 2        1   2   4
 5        5   2   4

PAGE FAULTS: 5
```