Ex. No. 11(a)	FIFO PAGE REPLACEMENT
Date: 04.04.2025	

Aim:

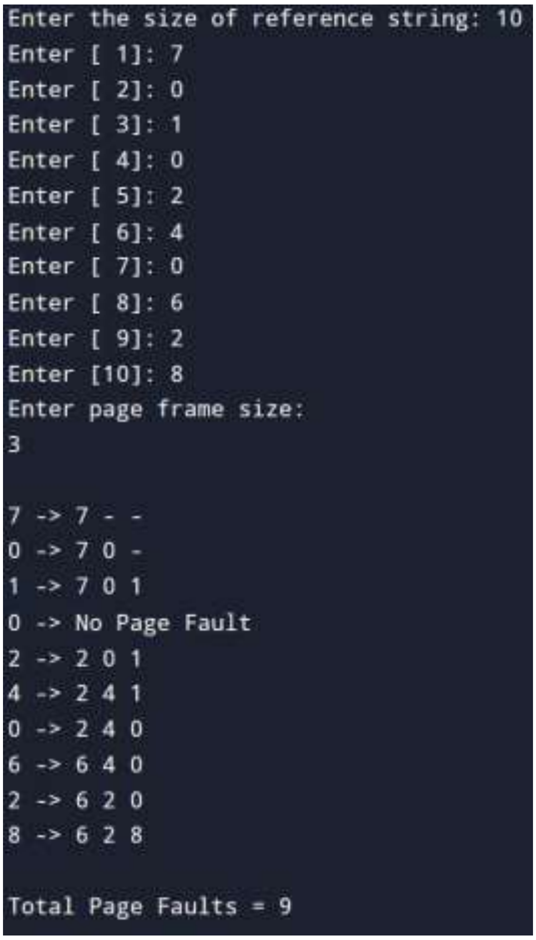
To find out the number of page faults that occur using First-in First-out (FIFO) page replacement technique.

Program:

```
#include <stdio.h> int
main() {
  int referenceString[50], page[20], frames, refLen, i, j, k, avail, pageFaults = 0,
next = 0;     printf("Enter the size of reference string: ");     scanf("%d",
&refLen);

  for (i = 0; i < refLen; i++) {
printf("Enter [%d] : ", i + 1);
    scanf("%d", &referenceString[i]);
  }
  printf("Enter page frame size: ");
scanf("%d", &frames);

  for (i = 0; i < frames; i++)
    page[i] = -1;

  for (i = 0; i < refLen; i++) {
avail = 0;
    for (j = 0; j < frames; j++) {            if
(page[j] == referenceString[i]) {
avail = 1;
        break;
      }
    }
    if (avail == 0) {
      page[next] = referenceString[i];
next   =   (next   +   1)   %   frames;
pageFaults++;

      for (k = 0; k < frames; k++)             page[k] != -1 ?
printf("%d ", page[k]) : printf("- ");           printf("-> Page
Fault\n");
```

```
    } else {
      for (k = 0; k < frames; k++)
        page[k] != -1 ? printf("%d ", page[k]) : printf("- ");
printf("-> No Page Fault\n");
    }
  }
  printf("Total Page Faults: %d\n", pageFaults);
return 0;
}
```

Output:



```
Enter the size of reference string: 10
Enter [ 1]: 7
Enter [ 2]: 0
Enter [ 3]: 1
Enter [ 4]: 0
Enter [ 5]: 2
Enter [ 6]: 4
Enter [ 7]: 0
Enter [ 8]: 6
Enter [ 9]: 2
Enter [10]: 8
Enter page frame size:
3

7 -> 7 - -
0 -> 7 0 -
1 -> 7 0 1
0 -> No Page Fault
2 -> 2 0 1
4 -> 2 4 1
0 -> 2 4 0
6 -> 6 4 0
2 -> 6 2 0
8 -> 6 2 8

Total Page Faults = 9
```

Result:

Thus, the program to implement FIFO Page Replacement was executed successfully and the number of page faults was determined correctly.

Ex. No. 11(b)	LRU PAGE REPLACEMENT
Date: 10.04.2025	

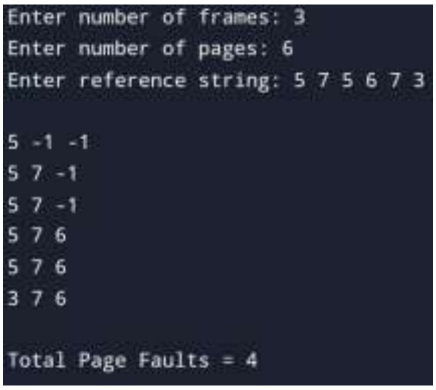
Aim:

To write a c program to implement LRU page replacement algorithm

Program:

```
#include <stdio.h> int main() {     int f[10], p[50], n,
m, i, j, k, pos, pf = 0, lru[10], least;

  printf("Enter number of frames: ");
scanf("%d", &n);

  printf("Enter number of pages: ");
scanf("%d", &m);

  printf("Enter reference string: ");
  for (i = 0; i < m; i++)
    scanf("%d", &p[i]);

  for (i = 0; i < n; i++) {
    f[i] = -1;
    lru[i] = 0;
  }

  printf("\n");

  for (i = 0; i < m; i++) {
    int found = 0;

    for (j = 0; j < n; j++) {
if (f[j] == p[i]) {
found = 1;               lru[j]
= i;
        break;
      }
    }

    if (!found) {
if (pf < n) {
f[pf] = p[i];
lru[pf] = i;            }
else {                least
= lru[0];
pos = 0;
for (j = 1; j < n;
j++) {                    if
(lru[j] < least) {
least = lru[j];
pos = j;
                    }
                }
                f[pos] = p[i];
                lru[pos] = i;
            }
pf++;
        }

        for (k = 0; k < n; k++) {            if
(f[k] != -1)  printf("%d ", f[k]);
else printf("-1 ");
        }
        printf("\n");
    }

    printf("\nTotal Page Faults = %d\n", pf);
return 0;
}
```

Output:



```
Enter number of frames: 3
Enter number of pages: 6
Enter reference string: 5 7 5 6 7 3

5 -1 -1
5 7 -1
5 7 -1
5 7 6
5 7 6
3 7 6

Total Page Faults = 4
```

Result:

Thus, the LRU Page Replacement Algorithm was successfully implemented, and the number of page faults was calculated based on the reference string.

Ex. No. 11(c)	**OPTIMAL PAGE REPLACEMENT**
Date: 11.04.2025	

Aim:

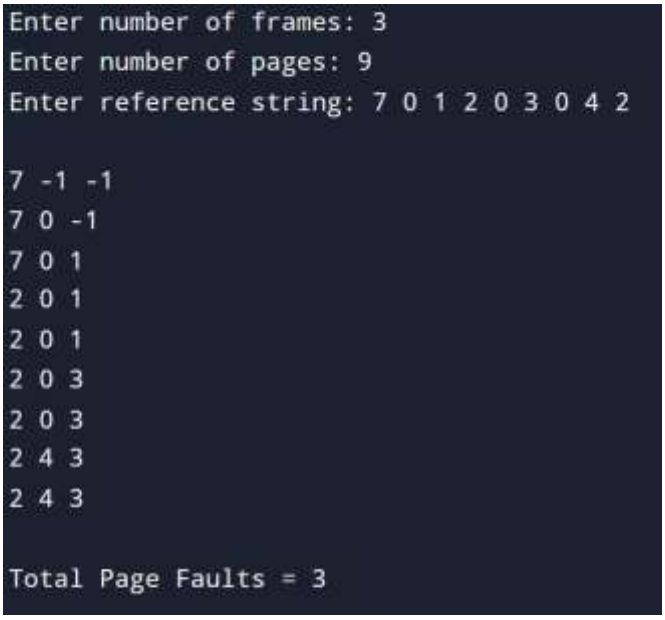
To write a c program to implement Optimal page replacement.

Program:

```
#include <stdio.h> int
main() {
   int f[10], p[50], i, j, k, pos, pf = 0, n, m, found, farthest, index;

   printf("Enter number of frames: ");
scanf("%d", &n);

   printf("Enter number of pages: ");
scanf("%d", &m);

   printf("Enter reference string: ");
   for (i = 0; i < m; i++)
      scanf("%d", &p[i]);

   for (i = 0; i < n; i++)
      f[i] = -1;

   printf("\n");

   for (i = 0; i < m; i++) {
      found = 0;

      for (j = 0; j < n; j++) {
if (f[j] == p[i]) {
found = 1;
            break;
         }
      }
```

```
    if (!found) {
if (pf < n) {
f[pf++] = p[i];
      } else {
        farthest = -1;         index
= -1;         for (j = 0; j < n; j++) {
int next = -1;           for (k = i + 1;
k < m; k++) {             if (f[j] ==
p[k]) {               next = k;
break;
            }
          }
          if (next == -1) {
index = j;
            break;
          } else if (next > farthest) {
farthest = next;             index =
j;
          }
        }
        f[index] = p[i];
      }
    }

    for (j = 0; j < n; j++) {
if (f[j] != -1)
printf("%d ", f[j]);
else
        printf("-1 ");
    }
    printf("\n");
  }

  printf("\nTotal Page Faults = %d\n", pf);
  return 0; }
```

Output:



```
Enter number of frames: 3
Enter number of pages: 9
Enter reference string: 7 0 1 2 0 3 0 4 2

7 -1 -1
7 0 -1
7 0 1
2 0 1
2 0 1
2 0 3
2 0 3
2 4 3
2 4 3

Total Page Faults = 3
```

Result:

Thus, the Optimal Page Replacement Algorithm was successfully implemented, and the number of page faults was calculated based on the reference string.