Task 11

Task 11a: To write C program to Simulate FIFO page replacement Algorithms for memory management:

Program:

```
#include <stdio.h>

int main() {

  int i, j, n, a[50], frame[10], no, k, avail, count = 0;

  // Input: Number of pages

  printf("\nENTER THE NUMBER OF PAGES:\n");

  scanf("%d", &n);

  // Input: Page reference string

  printf("\nENTER THE PAGE NUMBERS:\n");

  for (i = 1; i <= n; i++) {

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

  }

  // Input: Number of frames

  printf("\nENTER THE NUMBER OF FRAMES:\n");

  scanf("%d", &no);

  // Initialize frames to -1 (empty)

  for (i = 0; i < no; i++) {

    frame[i] = -1;

  }

  j = 0;

  printf("\nRef String\tPage Frames\n");

  // Loop through each page in reference string

  for (i = 1; i <= n; i++) {
```

```
printf("%d\t\t", a[i]);

        avail = 0;

        // Check if page is already in frame

        for (k = 0; k < no; k++) {

            if (frame[k] == a[i]) {

                avail = 1; // Page hit

                break;

            }

        }

        // If not available, it's a page fault

        if (avail == 0) {

            frame[j] = a[i];           // Replace using FIFO

            j = (j + 1) % no;          // Move to next frame index (circular)

            count++;                   // Increment page fault count

            // Print current frame content

            for (k = 0; k < no; k++) {

                printf("%d\t", frame[k]);

            }

        }

        printf("\n");

    }

    // Output the total number of page faults

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

    return 0;

}
```

Output:

```
ENTER THE NUMBER OF PAGES:
12

ENTER THE PAGE NUMBERS:
1 2 3 4 1 2 5 1 2 3 4 5

ENTER THE NUMBER OF FRAMES:
3
Ref String     Page Frames
1              1   -1  -1
2              1   2   -1
3              1   2   3
4              4   2   3
1              4   1   3
2              4   1   2
5              5   1   2
1
2
3              5   3   2
4              5   3   4
5              5   3   4

Total Page Faults = 9
```

Task 11b: To write C program to Simulate LRU page replacement Algorithms for memory management:

Program:

```
#include <stdio.h>


int main() {

   int frames[10], temp[10], pages[10];

   int total_pages, m, n, position = 0, k, l, total_frames;

   int a = 0, b = 0, page_fault = 0;


   // Input: Number of frames

   printf("\nEnter Total Number of Frames:\t");

   scanf("%d", &total_frames);


   // Initialize all frames to -1

   for (m = 0; m < total_frames; m++) {

      frames[m] = -1;

   }


   // Input: Number of pages

   printf("Enter Total Number of Pages:\t");

   scanf("%d", &total_pages);


   // Input: Reference string (page numbers)

   printf("Enter Values for Reference String:\n");

   for (m = 0; m < total_pages; m++) {
    printf("Value No.[%d]:\t", m + 1);
    scanf("%d", &pages[m]);
}

// Processing each page
for (n = 0; n < total_pages; n++) {
    a = 0, b = 0;

    // Check if the page is already in a frame
    for (m = 0; m < total_frames; m++) {
        if (frames[m] == pages[n]) {
            a = 1; // Page hit
            b = 1;
            break;
        }
    }

    // If not in frames, try to place in empty frame
    if (a == 0) {
        for (m = 0; m < total_frames; m++) {
            if (frames[m] == -1) {
                frames[m] = pages[n];
                b = 1;
                page_fault++;
                break;
        }
    }
}

// If no empty frame, use LRU logic to replace
if (b == 0) {
    for (m = 0; m < total_frames; m++) {
        temp[m] = 0;
    }

    // Look backward to find least recently used
    for (k = n - 1, l = 1; l <= total_frames - 1; l++, k--) {
        for (m = 0; m < total_frames; m++) {
            if (frames[m] == pages[k]) {
                temp[m] = 1;
            }
        }
    }

    // Find the frame which was not recently used
    for (m = 0; m < total_frames; m++) {
        if (temp[m] == 0) {
            position = m;
        }
    }
```

```
            // Replace the LRU page with current page
            frames[position] = pages[n];
            page_fault++;
        }

        // Print the current frame status
        printf("\n");
        for (m = 0; m < total_frames; m++) {
            printf("%d\t", frames[m]);
        }
    }

    // Final output
    printf("\n\nTotal Number of Page Faults:\t%d\n", page_fault);

    return 0;
}
```

Output:

```
Enter Total Number of Frames:   3
Enter Total Number of Pages:    12
Enter Values for Reference String:
Value No.[1]:   2
Value No.[2]:   3
Value No.[3]:   2
Value No.[4]:   1
Value No.[5]:   5
Value No.[6]:   2
Value No.[7]:   4
Value No.[8]:   5
Value No.[9]:   3
Value No.[10]:  2
Value No.[11]:  5
Value No.[12]:  2


2      -1      -1
2      3       -1
2      3       -1
2      3       1
2      5       1
2      5       1
2      5       4
2      5       4
3      5       4
3      5       2
3      5       2
3      5       2


Total Number of Page Faults:    7
```