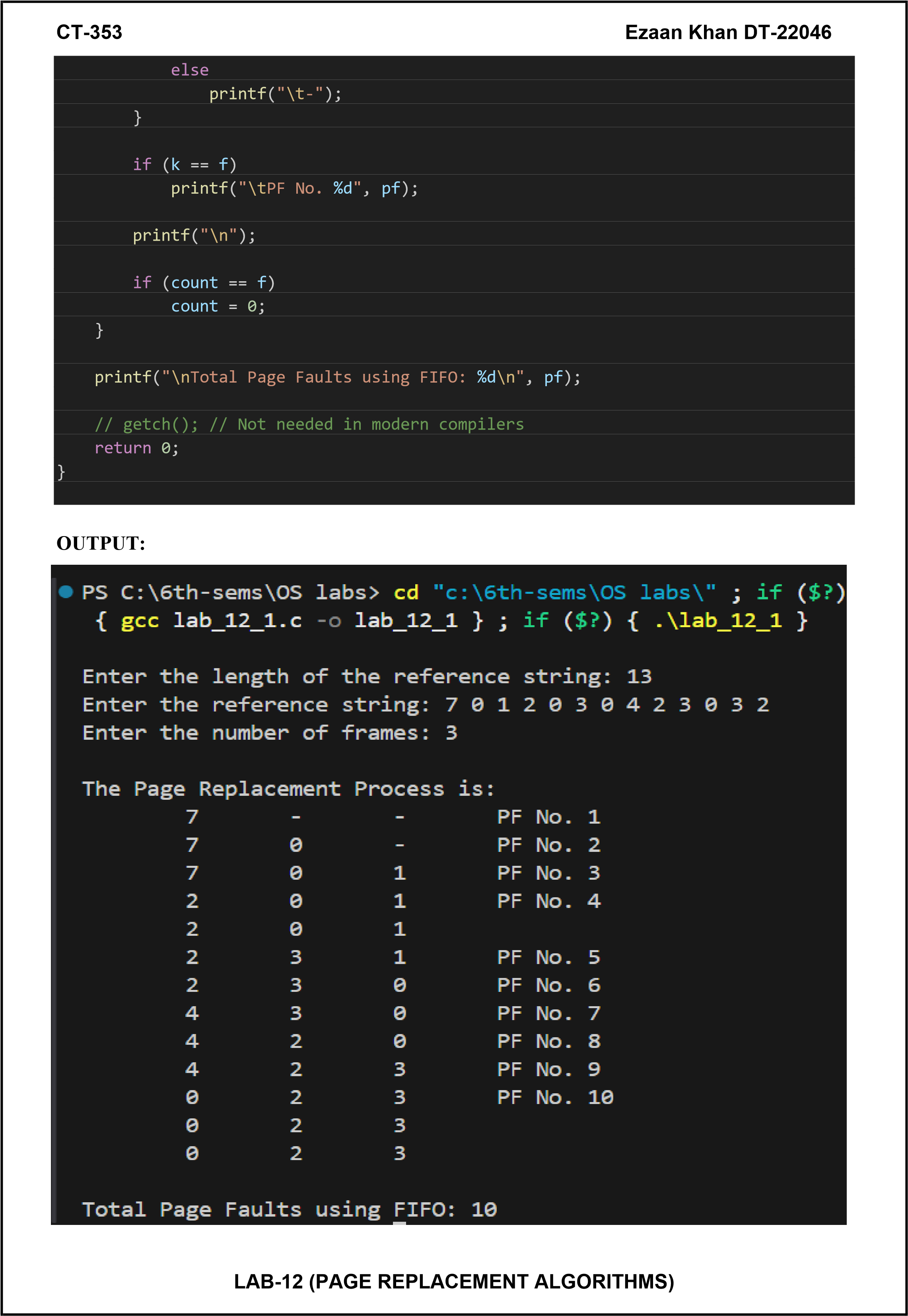
|  |
| --- |
| **CT-353 Talal Khan DT-22043**  **LAB-12**  **Exercise:**  1) Implement the above code and paste the screen shot of the output. **a) FIFO**  **PROGRAM:**  #include <stdio.h>  #include <stdlib.h> // for exit()  int main() { int i, j, k, f, pf = 0, count = 0; int rs[25], m[10], n;    // clrscr(); // Not needed in modern compilers  printf("\nEnter the length of the reference string: "); scanf("%d", &n);  printf("Enter the reference string: "); for (i = 0; i < n; i++) { scanf("%d", &rs[i]);  } printf("Enter the number of frames: "); scanf("%d", &f);  for (i = 0; i < f; i++) { m[i] = -1; // Initialize all frames to -1  } printf("\nThe Page Replacement Process is:\n");    for (i = 0; i < n; i++) { for (k = 0; k < f; k++) { if (m[k] == rs[i]) { break; // Page hit  }  }  if (k == f) { // Page fault m[count++] = rs[i]; pf++;  }    // Display current frame status for (j = 0; j < f; j++) { if (m[j] != -1) printf("\t%d", m[j]);  **LAB-12 (PAGE REPLACEMENT ALGORITHMS)** |

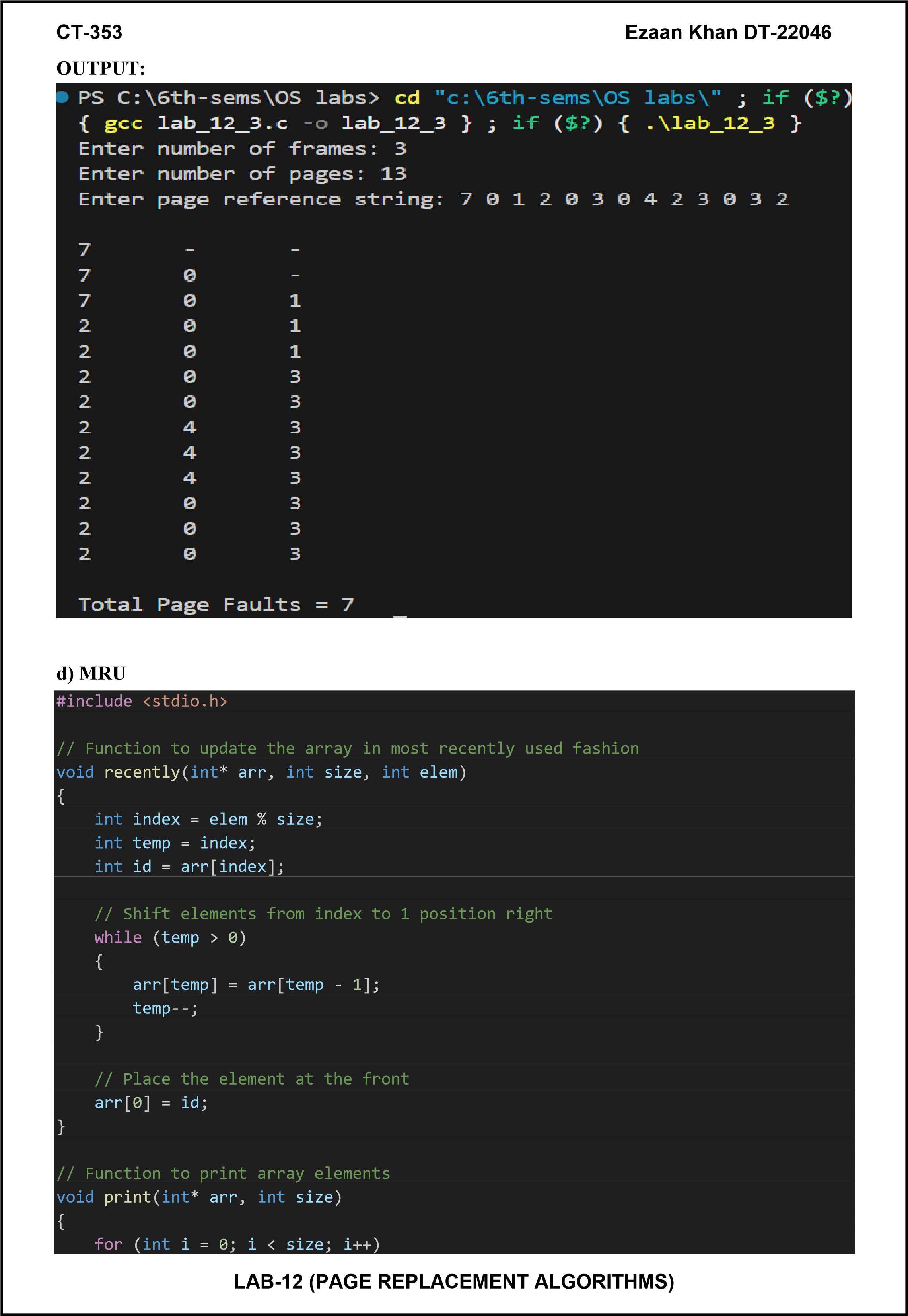


|  |
| --- |
| **CT-353 Ezaan Khan DT-22046**    **b) LRU**  #include <stdio.h>  #include <stdlib.h> // for exit()  int main() { int i, j, k, min, rs[25], m[10], count[10], flag[25]; int n, f, pf = 0, next = 1;    // clrscr(); // Not used in modern compilers  printf("Enter the length of the reference string: "); scanf("%d", &n);  printf("Enter the reference string: "); for (i = 0; i < n; i++) { scanf("%d", &rs[i]); flag[i] = 0;  } printf("Enter the number of frames: "); scanf("%d", &f);  for (i = 0; i < f; i++) { count[i] = 0; m[i] = -1;  } printf("\nThe Page Replacement Process is:\n");  for (i = 0; i < n; i++) { for (j = 0; j < f; j++) { if (m[j] == rs[i]) { flag[i] = 1; count[j] = next++; break;  }  }  if (flag[i] == 0) { if (i < f) { m[i] = rs[i]; count[i] = next++;  } else { min = 0;  for (j = 1; j < f; j++) { if (count[min] > count[j]) { min = j;  }  }  **LAB-12 (PAGE REPLACEMENT ALGORITHMS)** |



|  |
| --- |
| **CT-353 Ezaan Khan DT-22046 c) Optimal Page Replacement**  #include <stdio.h>  int main() { int no\_of\_frames, no\_of\_pages; int frames[10], pages[30], temp[10]; int flag1, flag2, flag3;  int 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;    // Check if page is already in a frame for (j = 0; j < no\_of\_frames; ++j) { if (frames[j] == pages[i]) { flag1 = flag2 = 1; break;  }  }    // If page is not in frame but there is empty space if (flag1 == 0) { for (j = 0; j < no\_of\_frames; ++j) { if (frames[j] == -1) { faults++;  frames[j] = pages[i]; flag2 = 1; break;  }  }  }    // If page is not in frame and no empty space, apply Optimal Replacement if (flag2 == 0) { flag3 = 0;  **LAB-12 (PAGE REPLACEMENT ALGORITHMS)** |

|  |
| --- |
| **CT-353 Ezaan Khan DT-22046**  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++;  }    // Print current state of memory frames printf("\n");  for (j = 0; j < no\_of\_frames; ++j) { if (frames[j] != -1) printf("%d\t", frames[j]); else  printf("-\t");  }  } printf("\n\nTotal Page Faults = %d\n", faults); return 0;  }      **LAB-12 (PAGE REPLACEMENT ALGORITHMS)** |



|  |  |
| --- | --- |
| **CT-353 Ezaan Khan DT-22046**   |  | | --- | | printf("%d ", arr[i]); printf("\n");  } int main() { int elem = 3; int arr[] = { 6, 1, 9, 5, 3 };  int size = sizeof(arr) / sizeof(arr[0]);  recently(arr, size, elem); printf("Array in most recently used fashion: "); print(arr, size);  return 0;  } |     **OUTPUT:**    **LAB-12 (PAGE REPLACEMENT ALGORITHMS)** |