Task 15: Write a program to calculate the number of page faults for a reference string using Optimal page replacement algorithm (input Takes by user).

Here's an example of the Optimal page replacement algorithm in C:

```
#include <stdio.h>
#include <limits.h>

#define SIZE 3 // Size of the page table

int isPageInMemory(int page, int pageTable[], int size) {
    for (int i = 0; i < size; i++) {
        if (pageTable[i] == page)
            return 1;
    }
    return 0;
}

int findOptimalPage(int referenceString[], int n, int pageTable[], int size, int index) {
    int res = -1, farthest = index;
    for (int i = 0; i < size; i++) {
        int j;
        for (j = index; j < n; j++) {
            if (pageTable[i] == referenceString[j]) {
                if (j > farthest) {
                    farthest = j;
                    res = i;
                }
                break;
            }
        }
```

```
        // If a page is not found in the future reference string, return it
        if (j == n)
            return i;
    }

    return (res == -1) ? 0 : res;
}

void optimalPageReplacement(int referenceString[], int n, int pageTable[], int size) {
    int pageFaults = 0;

    for (int i = 0; i < n; i++) {
        if (!isPageInMemory(referenceString[i], pageTable, size)) {
            int index = findOptimalPage(referenceString, n, pageTable, size, i);
            pageTable[index] = referenceString[i];
            pageFaults++;
        }

        // Display page table after each reference
        printf("Page Table: ");
        for (int j = 0; j < size; j++) {
            if (pageTable[j] == -1)
                printf("[ ] ");
            else
                printf("[%d] ", pageTable[j]);
        }
        printf("\n");
    }

    printf("Total Page Faults: %d\n", pageFaults);
```

```
}

int main() {
    int n;

    printf("Enter the number of page references: ");
    scanf("%d", &n);

    int referenceString[n];
    printf("Enter the page reference string:\n");

    for (int i = 0; i < n; i++)
        scanf("%d", &referenceString[i]);

    int pageTable[SIZE];
    for (int i = 0; i < SIZE; i++)
        pageTable[i] = -1;

    optimalPageReplacement(referenceString, n, pageTable, SIZE);

    return 0;
}
```

Output-

```
Enter the number of page references: 2
Enter the page reference string:
1
2
Page Table: [1] [ ] [ ]
Page Table: [2] [ ] [ ]
Total Page Faults: 2
```