#include<stdio.h>

#define MAX 3

typedef struct queue {

int aiframe[MAX];

} queue;

void fnOptimal(int arr[], int size) {

int pg = 0;

queue q;

for (int i = 0; i < MAX; i++) {

q.aiframe[i] = -1;

}

for (int i = 0; i < size; i++) {

int flag = 0;

for (int j = 0; j < MAX; j++) {

if (q.aiframe[j] == arr[i]) {

flag = 1;

break;

}

}

if (flag == 0) {

int replace\_index = -1;

int farthest = -1;

for (int j = 0; j < MAX; j++) {

int next\_use = -1;

for (int k = i + 1; k < size; k++) {

if (q.aiframe[j] == arr[k]) {

next\_use = k;

break;

}

}

if (next\_use == -1) {

replace\_index = j;

break;

} else if (next\_use > farthest) {

farthest = next\_use;

replace\_index = j;

}

}

if (replace\_index == -1) {

replace\_index = 0;

}

q.aiframe[replace\_index] = arr[i];

pg++;

}

// Print the current state of the frames

for (int j = 0; j < MAX; j++) {

if (q.aiframe[j] != -1) {

printf("%d ", q.aiframe[j]);

} else {

printf("- ");

}

}

printf("\n");

}

printf("\nTotal Page Faults: %d\n", pg);

}

int main() {

int size;

printf("Enter the size of reference string: ");

scanf("%d", &size);

int arr[size];

printf("Enter the reference String: ");

for (int i = 0; i < size; i++) {

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

}

fnOptimal(arr, size);

return 0;

}