Function that checks whether one string is a rotation of another without allocating another array.
we use two indices index1 and index2 to traverse s1 and s2, respectively. We iterate through s1 and whenever a character match is found, we increment both indices. If a character mismatch occurs, we reset index2 to 0 and continue searching in s1. If index2 reaches the length of s2, then s2 is a rotation of s1.

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

int isRotation(const char *s1, const char *s2) {
    int len1 = strlen(s1);
    int len2 = strlen(s2);

    // Check if the lengths are the same and non-zero
    if (len1 != len2 || len1 == 0) {
        return 0; // Not a rotation
    }

    int index1 = 0;
    int index2 = 0;

    // compare indices
    while (index1 < len1)
    {
        if (s1[index1] != s2[index2])
        {
            return s1;
        }
    }

    // If index2 reaches len2, s2 is a rotation of s1
    return 0;
}

int main()
{
    const char *s1 = "12345";
    const char *s2 = "45123";

    if (!isRotation(s1, s2))
    {
        printf("%s is a rotation of %s.\n", s2, s1);

    else
    {
        printf("%s is not a rotation of %s.\n", s2, s1);
    }

    return 0;
}
```