main.c

Share Run

Output

```c
//DAY 48
//Q95
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int areRotations(char *str1, char *str2) {
    int size1 = strlen(str1);
    int size2 = strlen(str2);

    if (size1 != size2) {
        return 0;
    }
    char *temp = (char *)malloc(sizeof(char) * (size1 * 2 + 1));
    temp[0] = '\0';
    strcat(temp, str1);
    strcat(temp, str1);
    int result = strstr(temp, str2) != NULL;
    free(temp);
    return result;
}
int main() {
    char str1[100], str2[100];

    printf("Enter first string: ");
    fgets(str1, sizeof(str1), stdin);
    printf("Enter second string: ");
    fgets(str2, sizeof(str2), stdin);

    str1[strcspn(str1, "\n")] = '\0';
    str2[strcspn(str2, "\n")] = '\0';

    if (areRotations(str1, str2)) {
        printf("Strings are rotations of each other.\n");
    } else {
        printf("Strings are not rotations of each other.\n");
    }
    return 0;
```

```
Enter first string: abcde
Enter second string: deabc
Strings are rotations of each other.


=== Code Execution Successful ===
```

main.c

Share Run

```c
//DAY 48
//Q96
#include <stdio.h>
void reverseWord(char *start, char *end) {
    char temp;
    while (start < end) {
        temp = *start;
        *start = *end;
        *end = temp;
        start++;
        end--;
    }
}
int main() {
    char str[100];
    int i = 0, start = 0;

    printf("Enter a sentence: ");
    fgets(str, sizeof(str), stdin);

    while (str[i] != '\0') {
        if (str[i] == ' ' || str[i] == '\n') {
            reverseWord(&str[start], &str[i - 1]);
            start = i + 1;
        }
        i++;
    }
    printf("Sentence with each word reversed: %s", str);
    return 0;
```

Output

```
Enter a sentence: I love coding
Sentence with each word reversed: I evol gnidoc


=== Code Execution Successful ===
```