

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define N 5
#define M 30

/* Κωδικας για διάταξη πίνακα δεικτών σε συμβολοσειρές
υλοποιημένες με δυναμική διαχείριση μνήμης (malloc).

ΔΙΝΕΤΑΙ ΜΟΝΟ Ο ΚΩΔΙΚΑΣ ΓΙΑ ΤΗΝ moveBubbleUp
την βασική συνάρτηση πρέπει να την υλοποιήσετε εσείς */

void moveBubbleUp(char * arr[], int n, int i);

/* Συνάρτηση για bubble sort σε πίνακα δεικτών σε strings που
χρησιμοποιεί επαναλαμβανόμενες κλήσεις της moveBubbleUp*/
void bubbleSortStringPointers(char *arr[], int n);

int main() {
    int i, len;
    char *arr[N]={0};
    // char x[N][M]={"else", "other", "another", "form", "that"};
    // Input strings
    printf("Enter the strings:\n");
    for ( i = 0; i < N; i++) {
        // arr[i]=x[i]
        char temp[M];
        /* printf("Give Length of word %d: ", i);
        scanf("%d", &len);*
        scanf("%s", temp);
        len=strlen(temp);
        arr[i]=(char *)malloc((len+1)*sizeof(char));
        strcpy(arr[i],temp);
        // arr[i]=temp; ΛΑΘΟΣ
    }

    // Κλήση συναρτησης διάταξης
    bubbleSortStringPointers(arr, N);
    // Εξαγωγή διατεταγμένων συμβολοσειρών
    printf("\nSorted strings:\n");
    for ( i = 0; i < N; i++) {
        printf("%s\n", arr[i]);

        // free(arr[i]); // Free allocated memory
    }

    return 0;
}

void moveBubbleUp(char * arr[], int n, int i)
{
    int j;

```

```

        char *temp;
        // Inner loop for comparing adjacent strings
        for (j = 0; j < n - i - 1; j++) {
            // Compare adjacent strings
            if (strcmp(arr[j], arr[j + 1]) > 0) {
                // Swap the strings if they are in the wrong order
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

```

```

/* Συνάρτηση για bubble sort σε πίνακα δεικτών σε strings */
void bubbleSortStringPointers(char *arr[], int n) {
    /* Γράψτε κώδικα που καλεί την moveBubbleUp(arr, n, i)
    και υλοποιεί τον bubblesort */

}

```