[**Task 1 (Pointers)**](http://testconfluence.training.local/pages/viewpage.action?pageId=14260911)

**Implement a dynamically sized list of c-strings.**

**Requirements:**

1) Only *stdlib.h* and *string.h*headers can be used in file where the list is implemented.

2) No *new*or *delete*operators. Only malloc, realloc and free functions.

3) Do not use classes or structures. Only pointers can be used.

4) You can use ANY code convention you like

5) User should be able to:

    - add/remove string

    - get the number of items in the list

    - get the index position of the first exact match of specified string

**Will be a plus:**

User is able to remove duplicates, replace the text in items, sort list.

**Example of interface:**

/\* Initializes list \*/

void StringListInit(char\*\*\* list);

/\* Destroy list and set pointer to NULL. \*/

void StringListDestroy(char\*\*\* list);

/\* Inserts value at the end of the list. \*/

void StringListAdd(char\*\* list, String str);

/\* Removes all occurrences of str in the list. \*/

void StringListRemove(char\*\* list, String str);

/\* Returns the number of items in the list. \*/

int StringListSize(char\*\* list);

/\* Returns the index position of the first occurrence of str in the list. \*/

int StringListIndexOf(char\*\* list, char\* str);

/\* Removes duplicate entries from the list. \*/

void StringListRemoveDuplicates(char\*\* list);

/\* Replaces every occurrence of the before, in each of the string lists's strings, with after. \*/

void StringListReplaceInStrings(char\*\* list, char\* before, char\* after);

/\* Sorts the list of strings in ascending order \*/

void StringListSort(char\*\* list);