#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main() {

FILE\* infile = fopen("strings.txt", "r");

char line[100];

int numStrings;

fgets(line, 100, infile);

sscanf(line, "%d", &numStrings);

char \*strings[numStrings];

char \*\*scan, \*\*endPoint;

int i;

// fill the array of strings with lines from the file:

for (i = 0; i < numStrings; i++) {

fgets(line, 100, infile);

strings[i] = malloc(strlen(line) + 1);

strcpy(strings[i], line);

}

// print out each element's value using indexing:

for (i = 0; i < numStrings; i++) {

printf("%s", strings[i] );

}

printf("\n");

// Alternatively, we could access each element and move // from one to the next with a pointer: \*\*scan

for (i = 0, scan = strings, endPoint = strings + numStrings;

scan < endPoint; i++, scan++)

printf("%s\tStack: %p, %p & Heap: %p, %p\n\n",

\*scan, &strings[i], scan, strings[i], \*scan);

// &strings[i] is address of the pointer, scan is the same-On the Stack!

// strings[i] is address of a string of char, \*scan is same-On the Heap!

// Simply show starting/ending addresses and the distance thereof:

printf("strings = %p, endPoint = %p\nstrings = %d, endPoint = %d\n"

"numStrings = %d\nTotal Pointers = %d\n"

"Actual distance of endPoint - strings = %d\n",

strings, endPoint, strings, endPoint,

numStrings, endPoint - strings,

(long)((long)endPoint - (long)strings));

// Best we clean up after ourselves:

fclose(infile);

for (i = 0, scan = strings; i < numStrings; i++, scan++) {

free(strings[i]);

strings[i] = NULL;

}

}

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main() {

FILE\* infile = fopen("strings.txt", "r");

char line[100];

char \*\*strings;

int numStrings, i;

fgets(line, 100, infile);

sscanf(line, "%d", &numStrings);

// Allocate enough memory to hold numStrings pointers to char

strings = (char \*\*) malloc(numStrings \* sizeof(char \*)); //cast doesn't matter.

// C99 can use the statement below (char \*strings[numStrings];) but,

// it should replace the line above and remove "char \*\*strings;" above.

// char \*strings[numStrings];

for (i = 0; i < numStrings; i++) {

fgets(line, 100, infile);

strings[i] = malloc(strlen(line) + 1); // Works C99 or earlier!

strcpy(strings[i], line);

printf("%s", strings[i] );

}

printf("\n");

for (i = 0; i < numStrings; i++) {

printf("%s", strings[i] );

}

fclose(infile);

// free all allocated memory:

for (i = 0; i < numStrings; i++) {

free(strings[i]);

strings[i] = NULL;

}

free(strings);

strings = NULL;

}