```c
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>

struct String {
    uint64_t length;
    char* contents;
};

typedef struct String String;

String new_String(char* init_contents) {
    uint64_t size = strlen(init_contents);
    char* contents = malloc(size + 1);
    strcpy(contents, init_contents);
    String r = { size, contents };
    return r;
}

// This plus is *just* the heap-allocating version now
String plus(String s1, String s2) {
    uint64_t new_size = s1.length + s2.length + 1;
    char* new_contents = calloc(new_size, sizeof(char));
    strncpy(new_contents, s1.contents, s1.length);
    strncpy(new_contents + s1.length, s2.contents, s2.length);
    new_contents[new_size - 1] = 0;
    String r = { new_size - 1, new_contents };
    return r;
}

String join(String strs[], int count, String delimiter) {
    String s = new_String("");
    for(int i = 0; i < count; i += 1) {
        s = plus(s, strs[i]);
        if(i < count - 1) {
            s = plus(s, delimiter);
        }
    }
    return s;
}

int main() {
    String apple = new_String("apple");
    String banana = new_String("banana");
    String strawberry = new_String("strawberry");
    String fruit[] = { apple, banana, strawberry };

    String comma = new_String(", ");
    String fruitlist = join(fruit, 3, comma);
    printf("%s\n", fruitlist.contents);
}
```

```
$ valgrind --leak-check=full ./jstr
==5777== Memcheck, a memory error detector
==5777== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==5777== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==5777== Command: ./jstr
==5777==
apple, banana, strawberry
==5777==
==5777== HEAP SUMMARY:
==5777==     in use at exit: 98 bytes in 10 blocks
==5777==   total heap usage: 11 allocs, 1 frees, 1,122 bytes allocated
==5777==
==5777== 1 bytes in 1 blocks are definitely lost in loss record 1 of 7
==5777==    at 0x483B7F3: malloc (in
    /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==5777==    by 0x109258: new_String (jstr.c:16)
==5777==    by 0x109368: join (jstr.c:34)
==5777==    by 0x109703: main (jstr.c:86)
==5777==
==5277== ... skip some from new_String ...
==5277==
==5777== 24 bytes in 2 blocks are definitely lost in loss record 6 of 7
==5777==    at 0x483DD99: calloc (in
    /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==5777==    by 0x1092D8: plus (jstr.c:25)
==5777==    by 0x1093D4: join (jstr.c:38)
==5777==    by 0x109703: main (jstr.c:86)
==5777==
==5777== 46 bytes in 3 blocks are definitely lost in loss record 7 of 7
==5777==    at 0x483DD99: calloc (in
    /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==5777==    by 0x1092D8: plus (jstr.c:25)
==5777==    by 0x1093A6: join (jstr.c:36)
==5777==    by 0x109703: main (jstr.c:86)
==5777==
==5777== LEAK SUMMARY:
==5777==    definitely lost: 98 bytes in 10 blocks
==5777==    indirectly lost: 0 bytes in 0 blocks
==5777==      possibly lost: 0 bytes in 0 blocks
==5777==    still reachable: 0 bytes in 0 blocks
==5777==         suppressed: 0 bytes in 0 blocks
==5777==
==5777== For lists of detected and suppressed errors, rerun with: -s
==5777== ERROR SUMMARY: 7 errors from 7 contexts (suppressed: 0 from 0)
```