Demonstration of unsafe string concatenation in C using `strcat()`

```
#include <stdio.h>
#include <string.h>

int main() {
   char s1[] = "security";
   char s2[15] = "software";
   char s3[10] = "";

   printf("Before strcat operations...\n");
   printf("s1: \"%s\" size: %zu length: %zu\n", s1, sizeof(s1), strlen(s1));
   printf("s2: \"%s\" size: %zu length: %zu\n", s2, sizeof(s2), strlen(s2));
   printf("s3: \"%s\" size: %zu length: %zu\n", s3, sizeof(s3), strlen(s3));

   strcat(s2,s1);          // append s1 to s2
   strcat(s3,s1);          // append s1 to s3
   strcat(s3,s2);          // append s2 to s3

   printf("After strcat() operations...\n");
   printf("s1: \"%s\" size: %zu length: %zu\n", s1, sizeof(s1), strlen(s1));
   printf("s2: \"%s\" size: %zu length: %zu\n", s2, sizeof(s2), strlen(s2));
   printf("s3: \"%s\" size: %zu length: %zu\n", s3, sizeof(s3), strlen(s3));
}
```

Let's assume the storage for arrays s1, s2, and s3 is allocated in the order of declarations in the source code (high memory address to low memory address).

Step 1: Show the stack frame contents BEFORE call to `strcat` function (left table on next page)

Step 2: Use the before call stack frame contents to determine the output produced by the first three print statements.

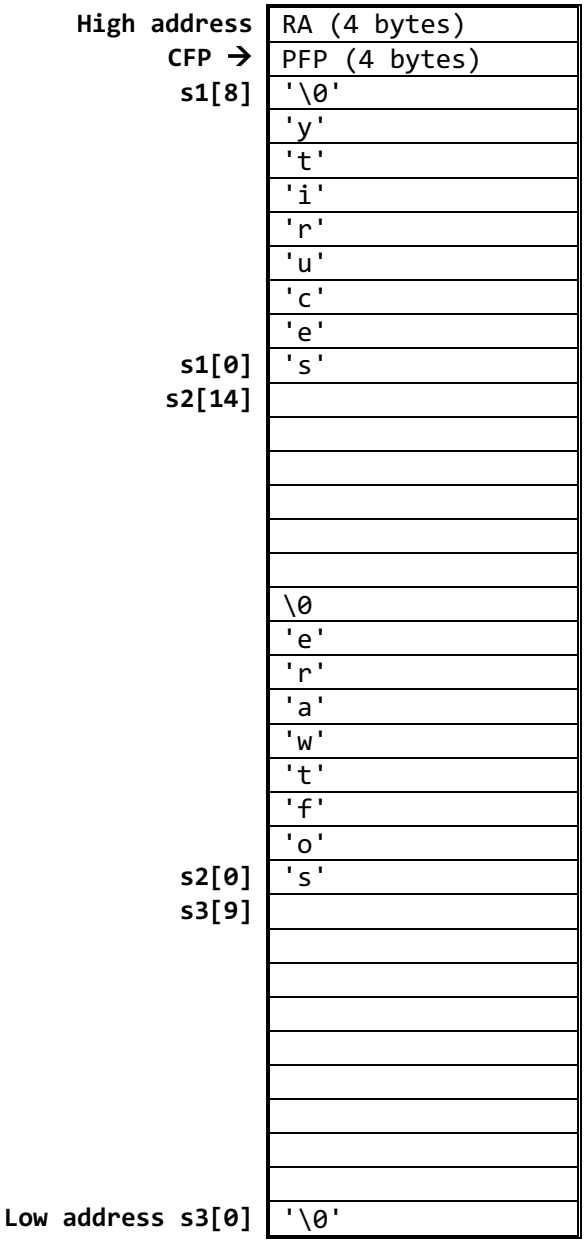
```
s1: "security" size: 9 length: 8
s2: "software" size: 15 length: 8
s3: "" size: 10 length: 0
```

Step 3: Show the stack frame contents AFTER call to `strcat` function (right table on next page)

Step 4: Use the after call stack frame contents to determine the output produced by the last three print statements.

```
s1: "y" size: 9 length: 1
s2: "ecurity" size: 15 length: 7
s3: "ysoftwaresecurity" size: 10 length: 17
```

Stack frame (before call to `strcat`)



Stack frame (after call to `strcat`)