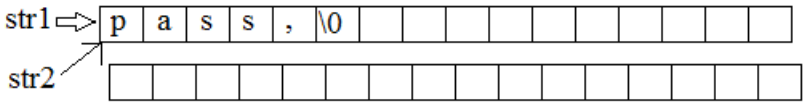
1) (5 pts) DSN (Dynamic Memory Management in C)

There is something terribly wrong with the code given below: it has two memory leaks. After carefully inspecting the code, answer the questions below.

```
1:    int main(void)
2:    {
3:      char *str1 = malloc(sizeof(char) * 16);
4:      char *str2 = malloc(sizeof(char) * 16);
5:
6:      str1[0] = 'p';
7:      str1[1] = 'a';
8:      str1[2] = 's';
9:      str1[3] = 's';
10:     str1[4] = ',';
11:     str1[5] = '\0';
12:
13:     printf("%s ", str1);
14:     str2 = str1;
15:     printf("%s ", str2);
16:     str2 = NULL;
17:     strcpy(str1, "pass the exam!");
18:     printf("%s\n", str1);
19:
20:     free(str1);
21:     free(str2);
22:
23:     return 0;
24:   }
```

(a) (3 pts) Draw a picture that indicates the relevant state of memory ***after*** line 14 has completed. (Draw a rectangular box to indicate dynamically allocated memory.)



Note: All cells left empty represent uninitialized character variables.

Grading: 1 pt for having two boxes drawn indicating allocated memory. 1 pt for having str1 point to the box that stores "pass," (this must be indicated), 1 pt for having str2 point to this same box.

(b) (1 pt) Explain why line 14 causes a memory leak.

When the pointer str2 moves, nothing is pointing to the memory that it used to be pointing to originally.

(c) (1 pt) Why is it possible for the code to crash on line 21?

str2 is pointing to NULL (nothing), so it's not pointing to dynamically allocated memory. Attempting to free memory that isn't dynamically allocated may crash a program.

Grading parts (b) and (c): Give the point for each if the answer is reasonably close or shows that the student understands the key issue at hand. **No half points! Only award an integer number of points.**