

1. In your own words, define array (buffer).

2. Examine the *char\_array.c* source code.

A) How does a program know it has reached the end of a string?

B) How many more characters could legitimately fit into this particular string?

3. A) If I have a source file name *'recipe.c'* and I want to turn it into an executable program named *'cake.exe'*, what command would I enter to compile this?

>

B) Now, what command would I enter to execute this program?

>

4. Given the string declaration below, mark each strcpy() function call as Valid (V) or Invalid (I).

```
char State[12];
```

A) `strcpy(State, "Oregon\n");` \_\_\_\_\_

B) `strcpy(State, "Detroit\n");` \_\_\_\_\_

C) `strcpy(State, "New Hampshire\n");` \_\_\_\_\_

D) `strcpy(State, "tj04wbn\n");` \_\_\_\_\_

E) `strcpy(State, "Mississippi\n");` \_\_\_\_\_

5. A) What is the name of the method that allows me to represent negative numbers in binary?

B) How would I recognize one of these negative binary numbers?

6. Study the OUTPUT of data type sizes in the middle of page 43. (Recall: 1 byte = 8 bits.) How many unique characters could I represent based on the size allocated for a char?