Question 1 (1 point)

Which answer most closely represents the values in `str` after this line executes:

```
char str[5];
```

NOTE: the - depicts a garbage value

- ◯ `{-, -, -, -, -}`
- ◯ `{'\0', '\0', '\0', '\0'}`
- ◯ `{-, -, -, -, -}`
- ◯ `{-, -, -, -}`
- ◯ `{0, 0, 0, 0}`
- ◯ `{'\0', '\0', '\0', '\0', '\0', '\0'}`
- ◯ `{0, 0, 0, 0, 0}`
- ◯ `{'\0', '\0', '\0', '\0', '\0'}`
- ◯ `{0, 0, 0, 0, 0, 0}`
- ◯ Not a valid declaration

Question 2 (1 point)

Which answer most closely represents the values in `str` after this line executes:

```
char str[];
```

NOTE: the - depicts a garbage value

- ◯ Not a valid declaration
- ◯ `{-}`
- ◯ `{}`
- ◯ `{0}`
- ◯ `{'\0'}`

▷ View hint for Question 2

Question 3 (1 point)

Which answer most closely represents the values in `str` after this line executes:

```
char str[5] = "Hi";
```

NOTE: the - depicts a garbage value

- ◯ `{'H', 'i', '\0', '\0'}`
- ◯ `{'H', 'i', '\0', -, -}`
- ◯ `{'H', 'i', -, -, -}`
- ◯ Not a valid declaration and assignment
- ◯ `{'H', 'i', '\0', -, -, -}`
- ◯ `{'H', 'i'}`
- ◯ `{'H', 'i', '\0'}`
- ◯ `{'H', 'i', '\0', '\0', '\0'}`
- ◯ `{'H', 'i', '\0', '\0', '\0', '\0'}`

Question 4 (1 point)

Which answer most closely represents the values in `str` after this line executes:

```
char str[] = "Hi";
```

NOTE: the - depicts a garbage value

- ◯ {'H', 'i', '\0', -, -}
- ◯ {'H', 'i', '\0', '\0'}
- ◯ {'H', 'i', '\0', '\0', '\0'}
- ◯ {'H', 'i', '\0', '\0', '\0', '\0'}
- ◯ {'H', 'i'}
- ◯ Not a valid declaration and assignment
- ◯ {'H', 'i', '\0'}
- ◯ {'H', 'i', '\0', -, -, -}
- ◯ {'H', 'i', -}

Question 5 (1 point)

Which answer most closely represents the values in `str` after this line executes:

```
char str[2] = "Hi";
```

NOTE: the - depicts a garbage value

- ◯ `{'H', 'i', '\0', '\0', '\0', '\0'}`
- ◯ `{'H', 'i', -, -, -}`
- ◯ `{'H', 'i', '\0', -, -, -}`
- ◯ `{'H', 'i', '\0'}`
- ◯ `{'H', 'i', '\0', '\0', '\0'}`
- ◯ `{'H', 'i', '\0', '\0'}`
- ◯ `{'H', 'i', '\0', -, -}`
- ◯ `{'H', 'i'}`

▷ View hint for Question 5

Question 6 (1 point)

What is the output when this code snippet executes:

```
char str[5] = "Hi";

printf("%s", str);
```

Question 7 (1 point)

What is the output of the following code snippet:

```
char str[5] = "HiHi!";
printf("%c", str[2]);
```

Question 8 (1 point)

What is the output of the following code snippet:

```
char str[5] = "HiHi!";
```

```
str[2] = '\0';

printf("%s", str);
```

View hint for Question 8

Question 9 (1 point)

Assuming the memory for str_1 and str_2 are allocated in contiguous memory locations, what is the output of the following code snippet:

```
char str_1[5] = "HiHi!";
char str_2[5] = "Bye";
str_1[4] = '\0';
printf("%s, %s", str_1, str_2);
```

Question 10 (1 point)

Assuming the memory for str_1 and str_2 are allocated in contiguous memory locations, what is the output of the following code snippet:

```
char str_1[5] = "HiHi!";
char str_2[5] = "Bye";
printf("%s, %s", str_1, str_2);
```

View hint for Question 10

Question 11 (1 point)

Assuming `ctype.h` header file was included, and given the following function definition:

```
void foo(char str[]) {
    while(*str != '\0') {
        if (isupper(*str)) {
            printf("%c", tolower(*str));
        } else{
            printf("%c", *str);
        }
        str++;
    }
}
```

What would the output of the following code snippet be:

```
char my_string[] = "Hello There!";
foo(my_string);
```

Question 12 (1 point)

Assuming `ctype.h` header file was included, and given the following function definition:

```
void foo(char str[]) {
    while(*str != '\0') {
        if (isupper(*str)) {
            tolower(*str);
        }
        str++;
    }
}
```

After the following code snippet executes:

```
char my_string[] = "Hello There!";
foo(my_string);
```

the value of `my_string` would be represented as:

- ◯ `{'H', 'e', 'l', 'l', 'o', ' ', 'T', 'h', 'e', 'r', 'e', '!', '\0'}`
- ◯ `{'h', 'e', 'l', 'l', 'o', ' ', 't', 'h', 'e', 'r', 'e', '!', '\0'}`
- ◯ `{'h', 'e', 'l', 'l', 'o', ' ', 't', 'h', 'e', 'r', 'e', '!'}`
- ◯ `{'H', 'e', 'l', 'l', 'o', ' ', 'T', 'h', 'e', 'r', 'e', '!'}`

Question 13 (1 point)

Assuming `ctype.h` header file was included, and given the following function definition:

```
void foo(char str[]) {
    while(*str != '\0') {
        if (isupper(*str)) {
            *str = tolower(*str);
        }
        str++;
    }
}
```

After the following code snippet executes:

```
char my_string[] = "Hello There!";
foo(my_string);
```

the value of `my_string` would be represented as:

- `{'h', 'e', 'l', 'l', 'o', ' ', 't', 'h', 'e', 'r', 'e', '!'}`
- `{'H', 'e', 'l', 'l', 'o', ' ', 'T', 'h', 'e', 'r', 'e', '!'}`
- `{'h', 'e', 'l', 'l', 'o', ' ', 't', 'h', 'e', 'r', 'e', '!', '\0'}`
- `{'H', 'e', 'l', 'l', 'o', ' ', 'T', 'h', 'e', 'r', 'e', '!', '\0'}`

Question 14 (1 point)

Assuming `string.h` and `ctype.h` header files were included, and given the following function definition:

```
void foo(char str1[], char str2[]) {
    int cmp = strcmp(str1, str2);
    if ( cmp == 0) {
        printf("A");
    } else if ( cmp < 0) {
        printf("B");
    } else {
        printf("C");
    }
}
```

What is the output when the following code snippet executes:

```
char s1[] = "ABBA";
char s2[] = "ABBA";
foo(s1, s2);
```

Question 15 (1 point)

Assuming `string.h` and `ctype.h` header files were included, and given the following function definition:

```
void foo(char str1[], char str2[]) {
    int cmp = strcmp(str1, str2);
    if ( cmp == 0) {
        printf("A");
    } else if ( cmp < 0) {
        printf("B");
    } else {
        printf("C");
    }
}
```

What is the output when the following code snippet executes:

```
char s1[] = "Abba";
char s2[] = "abba";
foo(s1, s2);
```

View hint for Question 15

Question 16 (1 point)

Assuming `string.h` and `ctype.h` header files were included, and given the following function definition:

```
void foo(char str1[], char str2[]) {
    int cmp = strcmp(str1, str2);
    if ( cmp == 0) {
        printf("A");
    } else if ( cmp < 0) {
        printf("B");
    } else {
        printf("C");
    }
}
```

What is the output when the following code snippet executes:

```
char s1[] = "snoop dog";
char s2[] = "abba";
foo(s1, s2);
```

View hint for Question 16