

ECE 3331, Dr. Hebert, Fall 2023 HW 06 due Friday 09/29

Problem 1. Section 6.2, Exercise 5.

5. Suppose that the variable `letters[0]` has the character `A` as its contents, and `letters[25]` has the character `Z` as its contents. Also assume that the variable `temp` is of type `char`. What results from the following slice of code?

```
temp = letters[ 25 ];
letters[ 25 ] = letters[ 0 ];
letters[ 0 ] = temp;
```

Problem 2. Chapter 6.2, Exercise 11.

11. What is the error in the following definition statement?

```
/* mixed has some char and some int variables */
int char mixed[ 100 ];
```

Problem 3. Chapter 6.2, Exercises 23.

23. Suppose that `a` is an array of type `int` and that each cell in the array holds either 1 or 0. Suppose also that `i` and `j` are valid indexes into `a`. Do the two fragments always produce the same output? Explain.

Problem 4. Section 6.4, Exercise 3.

3. What is printed?

```
char c[ 2 ] = "A";
printf( "%c\n", c[ 0 ] );
printf( "%s", c );
```

Problem 5. Section 6.4, Exercise 5.

5. What is printed?

```
printf( "%s\n", &stooge2[ 1 ] );
printf( "%c", stooge2[ 1 ] );
```

Problem 6. Section 6.4, Exercise 6.

6. Explain the difference between the two uses of `scanf`.

```
scanf( "%c", &stooge3[ 0 ] );
scanf( "%s", &stooge3[ 0 ] );
```

Problem 7. Section 6.4, Exercise 7, 9, and 11.

Exercises 7 through 15 assume the definition

```
char s[] = "Alfred Hitchcock";
```

7. How many cells are in the array `s`?

Which of Exercises 8 through 15 are likely errors? Explain. If the expression is acceptable, what will be printed?

8. `printf("%s", s[2]);`

9. `printf("%s", &s[2]);`

10. `printf("%s", s);`

11. `printf("%s", &s);`

Problem 8. Section 6.6, Exercise 5.

5. What is printed?

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

main()
{
    char str1[ 20 ] = "Far Out ";
    char str2[ 20 ] = "";

    strcat( str1, str2 );
    printf( "%s\n%s", str1, str2 );
    return EXIT_SUCCESS;
}
```

Problem 9. Section 6.6, Exercise 7.

7. What is printed?

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

main()
{
    char str1[] = "Great Deals";
    char str2[] = "Shady Lane";
    printf( "%s", strncpy( str1, str2, 5 ) );
    return EXIT_SUCCESS;
}
```

Problem 10. Section 6.6, Exercise 11.

11. What is printed?

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

main()
{
    char str[] = "My Brother Was an Only Child";
    printf( "%d\n", strlen( str ) );
    printf( "%d", strlen( &str[ 5 ] ) );
    return EXIT_SUCCESS;
}
```

Problem 11. Section 6.7, Exercise 1.

```
int length( char string[ ] )
{
    int count;    /* string's length */
```

1. Write an alternative version of the function **length**, using a **while** loop instead of a **for** loop.

Problem 12. Chapter 6.8, Exercise 7 .

7. Suppose that we define

```
int numbs[ 100 ][ 100 ]; /* array of 10,000 ints */
```

and store in each cell the sum of the two indexes that reference the cell. For example, the contents of **numbs**[5][87] would be 92. Now assume that we map **numbs** into a one-dimensional array **new_numbs**, which also has 10,000 integer variables, in such a way that each cell in **new_numbs** has the same contents as the corresponding cell in **numbs**. What is printed?

```
printf( "%d", new_numbs[ 67 ] );
printf( "%d", new_numbs[ 0 ] );
printf( "%d", new_numbs[ 876 ] );
printf( "%d", new_numbs[ 777 ] );
printf( "%d", new_numbs[ 2 ] );
```

Problem 13. Chapter 6.8, Exercise 8.

8. Find the index of the cell in **new_job_table** that corresponds to the cell **job_table**[**i**][**j**] (see Figure 6.21).

TWO-DIMENSIONAL		ONE-DIMENSIONAL
<code>job_table[0][0]</code>	\longleftrightarrow	<code>new_job_table[0]</code>
<code>job_table[0][1]</code>	\longleftrightarrow	<code>new_job_table[1]</code>
<code>job_table[0][2]</code>	\longleftrightarrow	<code>new_job_table[2]</code>
<code>job_table[0][3]</code>	\longleftrightarrow	<code>new_job_table[3]</code>
<code>job_table[1][0]</code>	\longleftrightarrow	<code>new_job_table[4]</code>
<code>...</code>		<code>...</code>
<code>job_table[99][0]</code>	\longleftrightarrow	<code>new_job_table[396]</code>
<code>job_table[99][1]</code>	\longleftrightarrow	<code>new_job_table[397]</code>
<code>job_table[99][2]</code>	\longleftrightarrow	<code>new_job_table[398]</code>
<code>job_table[99][3]</code>	\longleftrightarrow	<code>new_job_table[399]</code>

FIGURE 6.21 Representing a two-dimensional array as a one-dimensional array.

Problem 14. Chapter 6 Programming Exercises 6.1

- 6.1. Write a program with an array **numbs** defined and initialized as

```
int numbs[ 100 ] = { 1, 3, 5 };
```

Although **numbs** can hold 100 integers, only three cells have been initialized. Your program is to store integers in the remaining cells according to the formula

$$\text{numbs}[i] = \text{numbs}[0] + \text{numbs}[1] + \dots + \text{numbs}[i - 1]$$

for $i = 3, \dots, 99$. For example

```
numbs[ 3 ] = numbs[ 0 ] + numbs[ 1 ] + numbs[ 2 ] = 1 + 3 + 5 = 9,
numbs[ 4 ] = numbs[ 0 ] + numbs[ 1 ] + numbs[ 2 ] + numbs[ 3 ]
           = 1 + 3 + 5 + 9 = 18
```

After computing the values, print the array.