Answers

Practice Problems Week 7-8

These are additional practice problems (and therefore **completely OPTIONAL**) that the ULAs created for you to practice your skills and get as comfortable as possible with C.

For all questions, assume all necessary libraries are included, and all function calls are properly defined and the same ones you have seen in class, such as `remove_crlf`.

> If you have any doubts, please discuss in the Discord, a study group, or ULA office hours. **Do not email Dr. Gerber, as these are supplemental problems.**
>
> You will find the answer to these problems, including the challenge at the end of this page, in the Files section under Practice Problems. Or you can click here.

1. In how many lines will the print statement in line 6 output?

```c
1   int main(void)
2   {
3       char buffer[128];
4       printf("How is life like in Jupiter? ");
5       fgets(buffer, 127, stdin);
6       printf("The user wrote life is %s in Jupiter.\n", buffer);
7
8       return 0;
9   }
```

Since fgets adds the newline character at the end of user input and we don't remove it, line 6 will be split in two lines

2. In how many lines will the print statement in line 9 output?

```c
1   int main(void)
2   {
3       char buffer[128];
4
5       printf("How is life like in Jupiter? ");
6       fgets(buffer, 127, stdin);
7       remove_crlf(buffer);
8
9       printf("The user wrote life is %s in Jupiter.\n", buffer);
10
11      return 0;
12  }
```

Different from #1, we now use remove_crlf which removes the newline char from buffer, so line 9 will be printed out in a single line

3. What is the output if the user entered “not good”?

```
int main(void)
{
    char buffer[128];

    printf("How is life like in Jupiter? ");
    fgets(buffer, 127, stdin);
    remove_crlf(buffer);
    remove_crlf(buffer);
    remove_crlf(buffer);

    printf("The user wrote life is %s in Jupiter.\n", buffer);

    return 0;
}
```

→ remove newline character

→ do nothing since we already removed the newline character

→ do nothing since we already removed the newline character

The output is

The user wrote life is not good in Jupiter.

We will get user input, remove the newline character, and print it out.

If user entered "not good", then the buffer is

0	1	2	3	4	5	6	7	8	9
n	o	t		g	o	o	d	~~\n~~	\0

gets rewritten to \0

After remove_crlf:

0	1	2	3	4	5	6	7	8	9
n	o	t		g	o	o	d	\0	\0

4. What is the output if the user entered “3.14”?

```
void print_fav_num(void)
{
    char buffer[128];
    int fav_num;

    printf("What is your favorite number? ");
    fgets(buffer, 127, stdin);
    remove_crlf(buffer);
    fav_num = atoi(buffer);

    printf("Your number is %d!\n", fav_num);
}
```

not float!

Not atof!

3.14 will get truncated to 3, so the output is

Your number is 3!

5. How many times will the following loop run? (Hint: Don’t try to run it)

```
void the_coolest_loop(void)
{
    int i;

    for (i = 0; i < 10; i--)
    {
        printf("i is %d!\n", i);
    }
}
```

decreasing instead of increasing so i will always be less than 10

An "infinite" amount

6. What will the print statements in lines 9 and 17 output? Will it be the same? (You may use a calculator if you want to be exact.) Assume that uninitialized variables in C get assigned garbage values.

```
1   void print_sums(void)
2   {
3       int sum;   → gets garbage value
4
5       for (int j = 0; j != 10; j++)
6       {
7           sum = sum + j;   → keeps adding a
8       }                      random value with j
9       printf("Sum is %d\n", sum);
10      printf("Resetting...\n");
11
12      sum = 0;   → reset to 0
13      for (int j = 0; j != 10; j++)
14      {
15          sum = sum + j;   add 0+1+2+...+9
16      }
17      printf("Sum is now %d\n", sum);
18  }
```

Line 9 prints

Sum is ____ → a random value

Line 17 prints

Sum is 45

7. What will be the function output if the user runs the program two times. The first time, the user enters the “yes” and “179.67”. The second time, the user enters “no”.

```
void multiply(void)
{
    char buffer[128] = "100.23134";
    double num;

    printf("The number is %s! But sadly, it is a string.\n", buffer);
    printf("Converting the string number to a double...\n");
    num = atof(buffer);

    printf("It should now be a number, let's do some math!\n");
    printf("%lf * 71 = %lf\n", num, num * 71);
```

→ 100.23134 * 71 = 7116.42514

```
    printf("Would you like to multiply %lf by some number? ", num);
    // Here I go overwriting the buffer, oh no! What is going to happen?!
    fgets(buffer, 127, stdin);
    remove_crlf(buffer);

    printf("You entered %s\n", buffer);
```

→ You entered yes or You entered ____ → anything else

```
    if (strcasecmp(buffer, "yes") == 0) // User wants to multiply our number by something.
    {
        printf("Amazing! What do you want to print the number by? ");
        fgets(buffer, 127, stdin);
        remove_crlf(buffer);
        printf("Cool beans, you want to print it by %s. Let's do it.\n", buffer);

        printf("%lf * %s = %lf\n", num, buffer, num * atof(buffer));
    }
    else
    {
        printf(":( Fine.\n");
    }

    printf("I am exiting now...\n");
}
```

If user enters "yes" and "179.67"

100.23134 * 179.67 = 18008.56486

If user enters "no"

:(Fine. is printed

everything else prints the same

8. How many lines total will the following program print out?

```
void ignore_user_input(void)
{
    char buffer[128];
    printf("Hello.\nHow\nAre\nYou? ");
    fgets(buffer, 127, stdin);

    printf("Hello.\n");
    printf("How\n");
    printf("Are\n");
    printf("You? ");
    fgets(buffer, 127, stdin);

    printf("I don't really care, so goodbye!\n");
}
```

4

4

1

4+4+1 = 9 lines

9. In how many lines will the print statement in line 5 output?

```
1   void boring_print(void)
2   {
3       char buffer[128] = "I am a string\nth\nat\n\t\tis\nsplit weirdly...\n :S\n";
4
5       printf("%s", buffer);
6   }
```

1 + 1 + 1 + 1 + 1 + 1

= 6 lines

10. What will the following functions print out if the user enters "2.718" for both `print_num` and `print_another_num`?

```
void print_num(void)
{
    char buffer[128];
    float a;

    printf("What is a? ");
    fgets(buffer, 127, stdin);
    remove_crlf(buffer);

    printf("You said that a is %s\n", buffer);
    printf("I am going to set a equal to that\n");

    a = buffer;

    printf("a is %f\n", a);
}
```

Error! Trying to assign a variable from another type

Would not even allow the program to run

```
void print_another_num(void)
{
    char buffer[128];
    float a;

    printf("What is a? ");
    fgets(buffer, 127, stdin);
    remove_crlf(buffer);

    printf("You said that a is %s\n", buffer);
    printf("I am going to set a equal to that\n");

    a = atof(buffer);

    printf("a is %f\n", a);
}
```

Correct

Prints

a is 2.718

11. What will the following function output?

```c
void print_bob_name(void)
{
                            0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 → 16
    char buffer[128] = "My name is Bob\n";
    buffer[2] = '\0';

    printf("The string is %s\n", buffer);
}
```

The string is My

12. What does this function print?

```c
void funky_print(void)
{
    char buffer[128] = "How are you?";

    for (int i = 0; buffer[i] != '\0'; i++) // What is this doing?
    {
        printf("%c", buffer[i]); // omg a %c !!
    }
}
```

Prints the string character by character until it encounters the null terminator

Prints How are you?

13. Similar to the previous question, what does function print?

```c
void cool_print(void)
{
                            0 1 2 3 4 5 6 7 8 9 10 11 12 → 10
    char buffer[128] = "How are you?";
    int length = strlen(buffer);
                   = 12
                                   12
    for (int i = 0; i < length; i++) // What is this doing?
    {
        printf("%c", buffer[i]);
    }
}
```

→ get buffer string length

Prints string character by character until i reaches the string lentgth

Prints How are you?

14. What will main print in lines 10, 12, and 14?

```
1    void add_num(int x, int y)
2    {
3        x = x + y;
4    }
5
6    int main(void)
7    {
8        int x = 10;
9
10       printf("At first x was %d\n", x);
11       add_num(20, 10);
12       printf("Now x is %d\n", x);
13       add_num(x, 10);
14       printf("While x is now %d\n", x);
15
16       return 0;
17   }
```

At first x was 10

Now x is 10

While x is now 10

15. What will main print in lines 12, 14, and 16?

```
1    int x;
2
3    void add_num(int y)
4    {
5        x = x + y;
6    }
7
8    int main(void)
9    {
10       x = 10;
11
12       printf("At first x was %d\n", x);
13       add_num(20);
14       printf("Now x is %d\n", x);
15       add_num(10);
16       printf("While x is now %d\n", x);
17
18       return 0;
19   }
```

→ a wild global variable has appeared !

At first x was 10

Now x is 30

While x is now 40

Challenge
Create a program that asks the user for grades, prompting them to keep entering grades and saving them into an array. Once the user is done or the program reaches the maximum amount of grades that can be entered, print out the list of grades in the format below, along with the average.

As a requirement, make sure to include the following three function signatures:

- `int get_grades(float *grades, int n);`
  Prompt the user for a grade, save it to the `grades` array, then ask the user if they want to enter another grade. Repeat until a maximum amount of grades have been entered (however long you declare your array to be) or the user enters anything that is not "y" or "yes". **Return** the number of grades entered.

- `void print_grades(float *grades, int n);`
  Print all of the grades entered in a line, followed by a comma and space or a newline character, as necessary.

- `void average_grades(float *grades, int n);`
  Add all of the grades entered and divide it by the total amount of grades. Print the result.

You can create additional helper functions if you need them.

Example. If the user enters 7.3, 8.4, and 9.0. Your program will print the following:

```
You entered 3 grades. Here is a list of them:

7.30, 8.40, 9.00

Based on the previous grades, the average is: 8.23
```

As an extra challenge, find the maximum and minimum grades the user entered and print them.

Answer in files folder under Practice Problems