

CS 1313 010 Spring 2025 Homework #12

Quiz to be held in class 9:00-9:15am Monday Apr 14 2025

Please feel free to discuss these questions with your classmates, but **NOT** to copy each other.

NOTE: Except where and as explicitly permitted in writing (for example, in a Homework), you are **ABSOLUTELY FORBIDDEN** to **COPY EVEN A SINGLE CHARACTER** from,

or to have **ANY** shared code with, **ANY** other entity,
whether a human being (regardless of whether in CS1313 or not),
a text resource, a computing resource or anything else,
whether in person, on a local computer, online or anywhere else.

It's **INCREDIBLY EASY** for us to detect such copying, so **DON'T EVEN THINK ABOUT IT!**

1. Suppose that you are using the C standard math library function `sqrt` in a program that you are writing. **GIVE ALL OF THE PREPROCESSOR DIRECTIVE(S)** that are needed at the top of your source file, in the **CORRECT ORDER**.

2. Suppose that you are using the C standard math library function `sqrt` in a program that you are writing, and suppose that the name of your source file is `my_math.c` (and that therefore the name of your executable will be `my_math`). **GIVE THE COMPLETE MAKEFILE ENTRY** that would be associated with this program. (**DON'T** include the `clean` entry.)

3. Consider the program below. **WHAT IS THE OUTPUT** of the program for the following inputs? (You **DON'T** need to show the output of the prompt message.) If you are not confident of an answer, type in, compile and run the program.

```
#include <stdio.h>
#include <math.h>

int main ()
{ /* main */
    float input_value;

    printf("Input a real number:\n");
    scanf("%f", &input_value);
    printf("fabs(%5.2f)          = %5.2f\n",
           input_value, fabs(input_value));
    printf("sqrt(fabs(%5.2f))   = %5.2f\n",
           input_value, sqrt(fabs(input_value)));
    printf("rint(%5.2f)         = %5.2f\n",
           input_value, rint(input_value));
    printf("rint(fabs(%5.2f))   = %5.2f\n",
           input_value, rint(fabs(input_value)));
    printf("floor(%5.2f)        = %5.2f\n",
           input_value, floor(input_value));
    printf("floor(fabs(%5.2f)) = %5.2f\n",
           input_value, floor(fabs(input_value)));
    printf("ceil(%5.2f)         = %5.2f\n",
           input_value, ceil(input_value));
    printf("ceil(fabs(%5.2f))   = %5.2f\n",
           input_value, ceil(fabs(input_value)));
} /* main */
```

- (a) **6.75**
- (b) **-6.75**
- (c) **6.25**
- (d) **-6.25**

4. **WHAT IS THE OUTPUT** of each of these programs? If you aren't confident of your answer, type in, compile and run the program.

(a) #include <stdio.h>

```
int main ()
{ /* main */
    int f1;
    int func1();

    printf("Inside main\n");
    f1 = func1();
    printf("Inside main, f1 = %d\n", f1);
    return 0;
} /* main */

int func1 ()
{ /* func1 */
    printf("Inside func1\n");
    return 99;
} /* func1 */
```

(b) #include <stdio.h>

```
int main ()
{ /* main */
    int f2;
    int func2(int x);

    printf("Inside main\n");
    f2 = func2(99);
    printf("Inside main, f2 = %d\n", f2);
    return 0;
} /* main */

int func2 (int x)
{ /* func2 */
    printf("Inside func2, x = %d\n", x);
    return x;
} /* func2 */
```

(c) #include <stdio.h>

```
int main ()
{ /* main */
    int f3, t = 99;
    int func3(int x);

    printf("Inside main, t = %d\n", t);
    f3 = func3(t);
    printf("Inside main, f3 = %d\n", f3);
    return 0;
} /* main */

int func3 (int x)
{ /* func3 */
    printf("Inside func3, x = %d\n", x);
    return x;
} /* func3 */
```

(d) #include <stdio.h>

```
int main ()
{ /* main */
    int f4, x = 99;
    int func4(int x);

    printf("Inside main, x = %d\n", x);
    f4 = func4(x);
    printf("Inside main, f4 = %d\n", f4);
    return 0;
} /* main */

int func4 (int x)
{ /* func4 */
    printf("Inside func4, x = %d\n", x);
    return x;
} /* func4 */
```

(e) #include <stdio.h>

```
int main ()
{ /* main */
    int f5, x = 99, y = 77;
    int func5(int x);

    printf("Inside main, x = %d, y = %d\n", x, y);
    f5 = func5(x) + func5(y);
    printf("Inside main, f5 = %d\n", f5);
    return 0;
} /* main */

int func5 (int x)
{ /* func5 */
    printf("Inside func5, x = %d\n", x);
    return x;
} /* func5 */
```

(f) #include <stdio.h>

```
int main ()
{ /* main */
    int f6, x = 99;
    int func6(int x);

    printf("Inside main\n");
    f6 = func6(x);
    printf("Inside main, f6 = %d\n", f6);
    return 0;
} /* main */

int func6 (int x)
{ /* func6 */
    printf("Inside func6, x = %d\n", x);
    return x * x;
} /* func6 */
```

(g) #include <stdio.h>

```
int main ()
{ /* main */
    int f7, x = 99, y = 77;
    int func7(int x);

    printf("Inside main, x = %d, y = %d\n", x, y);
    f7 = func7(x) + func7(y);
    printf("Inside main, f7 = %d\n", f7);
    return 0;
} /* main */

int func7 (int x)
{ /* func7 */
    printf("Inside func7, x = %d\n", x);
    return x * x;
} /* func7 */
```

(h) #include <stdio.h>

```
int main ()
{ /* main */
    int f8x, f8y, x = 44, y = -33;
    int func8(int x);

    printf("Inside main, x = %d, y = %d\n", x, y);
    f8x = func8(x);
    f8y = func8(y);
    printf("Inside main, f8x = %d\n", f8x);
    printf("Inside main, f8y = %d\n", f8y);
    return 0;
} /* main */

int func8 (int x)
{ /* func8 */
    printf("Inside func8, x = %d\n", x);
    if (x < 0) {
        return 0;
    } /* if (x < 0) */
    else {
        return 1;
    } /* if (x < 0)...else */
} /* func8 */
```

(i) #include <stdio.h>

```
int main ()
{ /* main */
    int f9x, f9y, x = 44, y = -33;
    int func9(int x);

    printf("Inside main, x = %d, y = %d\n", x, y);
    f9x = func9(x);
    f9y = func9(y);
    printf("Inside main, f9x = %d\n", f9x);
    printf("Inside main, f9y = %d\n", f9y);
    return 0;
} /* main */

int func9 (int x)
{ /* func9 */
    int return_value;
    int i;

    return_value = 1;
    printf("Inside func9, x = %d\n", x);
    if (x < 1) {
        return_value = 0;
    } /* if (x < 1) */
    else {
        for (i = 1; i <= x; i++) {
            return_value *= i;
        } /* for i */
    } /* if (x < 1)...else */
    return return_value;
} /* func9 */
```

(j) For this item, you **DON'T** need to show the prompt or input.

Use the following input:

22 -11 22 -44 33 22 77 -11

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

int main ()
{ /* main */
    const int number_of_values = 8;
    int value[number_of_values];
    int f10x, f10y, x = 22, y = -11;
    int element;
    int func10(int* array, int array_length, int target);

    printf("Inside main, x = %d, y = %d\n", x, y);
    printf("What are the %d elements?\n", number_of_values);
    for (element = 0; element < number_of_values; element++) {
        scanf("%d", &value[element]);
    } /* for element */
    printf("Inside main, value = ");
    for (element = 0; element < number_of_values; element++) {
        printf("%d ", value[element]);
    } /* for element */
    printf("\n");
    f10x = func10(value, number_of_values, x);
    f10y = func10(value, number_of_values, y);
    printf("Inside main, f10x = %d\n", f10x);
    printf("Inside main, f10y = %d\n", f10y);
    return 0;
} /* main */

/* Function func10 is on the next page. */
```



```

int func10 (int* array, int array_length, int target)
{ /* func10 */
    const int initial_instances          = 0;
    const int first_element              = 0;
    const int minimum_number_of_elements = 1;
    const int program_failure_code       = -1;
    int number_of_instances;
    int element;

    if (array_length < minimum_number_of_elements) {
        printf("ERROR: can't have an array of %d elements.\n",
            array_length);
        exit(program_failure_code);
    } /* if (array_length < minimum_number_of_elements) */
    if (array == (int*)NULL) {
        printf("ERROR: can't have a nonexistent array ");
        printf("of %d elements.\n", array_length);
        exit(program_failure_code);
    } /* if (array == (int*)NULL) */
    number_of_instances = initial_instances;
    for (element = first_element;
        element < array_length; element++) {
        if (array[element] == target) {
            number_of_instances++;
        } /* if (array[element] == target) */
    } /* for element */
    return number_of_instances;
} /* func10 */

```

(k) For this item, you **DON'T** need to show the prompt or input.

Use the following input:

22 -11 22 -44 33 22 77 -11

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

int main ()
{ /* main */
    const int number_of_values = 8;
    int value[number_of_values];
    int f11, element;
    int func11(int* array, int array_length);

    printf("What are the %d elements?\n", number_of_values);
    for (element = 0; element < number_of_values; element++) {
        scanf("%d", &value[element]);
    } /* for element */
    printf("Inside main, value = ");
    for (element = 0; element < number_of_values; element++) {
        printf("%d ", value[element]);
    } /* for element */
    printf("\n");
    f11 = func11(value, number_of_values);
    printf("Inside main, f11 = %d\n", f11);
    return 0;
} /* main */

/* Function func11 is on the next page. */
```

```

int func11 (int* array, int array_length)
{ /* func11 */
    const int first_element          = 0;
    const int second_element         = 0;
    const int minimum_number_of_elements = 1;
    const int program_failure_code    = -1;
    int greatest_value, greatest_value_index;
    int element;

    if (array_length < minimum_number_of_elements) {
        printf("ERROR: can't have an array of %d elements.\n",
            array_length);
        exit(program_failure_code);
    } /* if (array_length < minimum_number_of_elements) */
    if (array == (int*)NULL) {
        printf("ERROR: can't have a nonexistent array ");
        printf("of %d elements.\n", array_length);
        exit(program_failure_code);
    } /* if (array == (int*)NULL) */
    greatest_value = array[first_element];
    greatest_value_index = first_element;
    if (array_length > minimum_number_of_elements) {
        for (element = second_element;
            element < array_length; element++) {
            if (array[element] > greatest_value) {
                greatest_value      = array[element];
                greatest_value_index = element;
            } /* if (array[element] > greatest_value) */
        } /* for element */
    } /* if (array_length > minimum_array_length) */
    return greatest_value_index;
} /* func11 */

```

If you use ANY resources other than Dr. Neeman, the TAs/graders (Basiri, Bilal), the course textbook or the materials posted on the course webpage, you MUST reference them on the quiz. **THIS INCLUDES CLASSMATES, FRIENDS, PROFESSORS, ONLINE RESOURCES, ETC.**