

CMSC106 Final Exam Practice Questions

These questions may help you prepare for the final exam. Solutions will not be provided; however, you are welcome to discuss your solutions with TAs and your instructor during office hours. Keep in mind that the material covered by the final exam does not need to be exactly the same material covered by these questions (do not assume that because a topic is not part of these review questions the topic will not appear in the final exam). Check the final exam posting for information regarding the topics associated with the exam.

1. What happens when the following code fragment is executed?

```
char *x = NULL;
*x = 10;
```

2. What happens when the following code fragment is executed?

```
char location[3];
strcpy(location, "Maryland");
```

3. Rewrite the following loop using a do while.

```
i = 0;
while(i < size) {
    printf("%d\n", i);
    i++;
}
```

4. Rewrite the following function using a switch statement instead of a cascading if statement.

```
void assign_teacher(int age, char teacher[]) {
    if (age == 10)
        strcpy(teacher, "Mrs. Smith");
    else if (age == 12)
        strcpy(teacher, "Mr. Peterson");
    else if (age == 9 || age == 8)
        strcpy(teacher, "Mrs. Sanders");
    else
        strcpy(teacher, "Mr. Roberts");
}
```

5. Write a function that reads a number and prints the even numbers between 1 and the value provided.
6. Write a function named `arrays_equals` that has the following specifications:

Prototype: `int arrays_equals(int a[], int b[], int size_a, int size_b)`

Parameter: `a` and `b` are one-dimensional arrays of integers

Processing: the function returns true if the two arrays have the same corresponding values. For example:

[10, 20, 345] is equal to [10, 20, 345]

[10, 20, 345] is not equal to [20, 10, 345]

[10, 20, 345] is not equal to [10, 20]

[10, 20, 345] is not equal to []

7. Write a function named **product** that has the following specifications:

Prototype: `float product(float a[], int size_a)`

Parameter: `a` is a one-dimensional array of floating-point values

Processing: the function returns the product of the values in the array

8. Write a function that counts how many vowels are in a string.
9. Write a function that determines whether a string is a palindrome.
10. The following structure will be used for the questions that follow:

```
#define MAX_LENGTH 80
#define MAX_NUMBER_OF_PHONES 10

typedef struct phone {
    int area_code;
    int three_digits;
    int four_digits;
} Phone;

typedef struct phonebook {
    char names[MAX_NUMBER_OF_PHONES][MAX_LENGTH + 1];
    Phone phones[MAX_NUMBER_OF_PHONES];
    int number_of_phones;
} Phonebook;
```

- a. Given the following two variables, which of the accesses below are valid?

```
Phonebook pbook;
Phonebook *ptr = &pbook;
```

For example, pbook.name is a valid access, but pbook->name is invalid.

- i. pbook.phones
- ii. ptr.phones
- iii. ptr->names
- iv. ptr->names[0]
- v. ptr->names[0][1]
- vi. ptr->phones[0].area_code
- vii. ptr->phones[0]->four_digits
- viii. ptr->phones[0]->names
- ix. (*ptr).phones[0]

- b. Define a function that initializes a Phone structure.
 - c. Define a function that prints a Phone structure.
 - d. Define a function called **init_phonebook** that initializes a Phonebook structure with 0 phones.
 - e. Define a function called **add_phone** that adds a phone to the Phonebook structure. When adding a phone, you need the name of a person and the phone to add. The index associated with a person's name in the names array will be the index to use to store the phone number in the phones array. So, after inserting the first phone, names[0] will have the person's name, and phones[0] the person's phone.
 - f. Define a function called **print_phonebook** that prints the name and phone of everyone in the phonebook.
 - g. Define a function called **remove_phone** that removes a phone from the Phonebook structure. To remove a Phone you will shift Phone structures in the array to the left.
11. Write a function that prints the contents of a two-dimensional array by printing the last row first. For example, if the array contents is:

```
10, 7, 40
8, 9, 10
```

The function will print the values:

```
8 9 10
10 7 40
```

12. Write a function that compares two two-dimensional arrays and returns true if the two arrays have the same elements and in the same positions. For example:

10, 7, 40	is equal to	10, 7, 40
8, 9, 10		8, 9, 10

10, 7, 40	is not equal to	10, 7, 40
8, 9, 10		9, 8, 10

10, 7, 40	is not equal to	10, 7, 40
8, 9		9, 8, 10

13. Write a function named **linearize** that has the following specifications:

Prototype: `void linearize(int data[2][3], int max_row, int max_col, int result[])`

Parameter: data is a two-dimensional array of integers

Processing: the function initializes result (one-dimensional array) with the elements of the two dimensional array. For example, if the two-dimensional array has the following values:

10, 7, 40
8, 9, 10

the array results will have the values:

10, 7, 40, 8, 9, 10