

CMSC 216 Quiz 3 Worksheet

The third quiz for the course will be on Fri, Jun 22. The following exercises cover the material to be included in this quiz. Solutions to these exercises will not be provided, but you are welcome to discuss your solutions with the TA or instructor during office hours.

Exercises

1. Define a function call maxAndMin that has the following prototype:

```
void maxAndMin(int a[], int a_size, int *max, int *min)
```

The function will find the maximum and minimum present in array a and return those values via the pointers parameters.

2. You need to implement the function **maximum** that has the prototype below. The function computes the maximum in the array and returns that value via the max parameter. If the array has a size of 0, the pointer variable associated with the argument must be set to NULL. The following code fragment illustrates how the function will be used.

```
int b[] = {30, 5, 80, 4};
int max;
int* maxPtr = &max;

maximum(b, 4, &maxPtr);
if (maxPtr == NULL) {
    printf("Array size is 0\n");
} else {
    printf("%d\n", max);
}
```

You can assume the array passed to the function will have positive elements (if the array size is different from 0).

```
void maximum(int a[], int a_size, int **max)
```

3. Define a method call intersection that has the following prototype:

```
int intersection(int *a, int size_of_a, int *b, size_of_b, int *common);
```

The method will initialize the common array with elements that are present in arrays a and b. The method will return the number of elements in common. You can assume the common array is large enough to fit the result.

4. Define a structure (using typedef) called Name that has the following fields:

```
first_name – maximum size 80
last_name – maximum size 80
```

5. Define a compareName method that takes two Name structures and returns -1 if first parameter precedes the second, 0 if they are equal and 1 otherwise.

6. Define a structure (using typedef) called Customer that has the following fields:

```
full_name – of type Name (defined above)
age → integer
phone → string (maximum size 12)
```

7. Define a method call printCustomer that prints the contents of a Customer structure.

8. Define a function called initialize that reads the full name, age, and phone number of a customer and returns a Customer structure initialized with those values. For example, the method will be called as follows: initialize("John", "Smith", 45, "301-555-5555").
9. Define a function called equals that takes two pointers to Customer structures and determines whether they are equal.
10. Define a program that using command line arguments displays a square root table from 1 up to a specified limit. For example, we can run the program as follows: a.out 100