HW #8: Arrays and Functions

Due dates:

Part I: Monday March 14th, at the beginning of the class. Make sure to write your name and msunetid on your paper.
Part II: Sunday March 13th, 11:59 pm through Handin (https://secure.cse.msu.edu/handin)

Part I: Comprehension Questions

1. Write a declaration of an array named calendar containing 12 integer values. Include an initializer that sets all 12 values: the first value to the number of days in January, the second to the number of days in February, … (1 pt)

```
int calendar[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
or
int calendar[ ] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

2. Declare an array of size 10 of type char. Write a loop that initializes each value of the array to a random character between 'A' and 'Z'. (2 pt)

```
char letters[10];
for (int index=0; index<10; index++) {
    letters[index] = 'A' + rand() % 26;
}
```

3. Declare an array of size 1000 of type int and initialize the first 3 elements to 1, 2, and 3 respectively and the remaining elements to 0 without using a loop. (1 pt)

```
int numbers[1000] = {1, 2, 3};
```

4. Fix four errors in the following program: (2 pts)

```
#include <stdio.h>
int main(void) {
     int[5] a = {10, 20, 20, 30, 50};      //Should be: int a[5] = {10, 20, 20, 30, 50};
     int b[ ] = {5, 6, 1, 4, 0};
     //copy array a into array b:
     b = a;                                //Cannot copy array with direct assignment.
                                           //Use:   memcpy(a, b, sizeof(a))
     //Multiply each element of b by 2
     for (x=0; x<=5; x++)                  //Wrong boundaries, undeclared x: for (int x=0; x<5; x++)
        b[x] = b[x*2];                     //Double the value not the index: b[x] = b[x] *2;
     return 0;
}
```

5. What is the output of the following program? (2 pts)

```
 #include <stdio.h>
    int jump(int pos, int steps) {
        return pos + steps;
    }
    int main(void) {
         int x = 6;
         int w = 3;
         printf("%d %d\n", jump(x + 3, 3), jump(jump(x+3, 3), 2) );
         return 0;
    }
```

12 14

jump(x+3, 3) ⇔ jump(6+3, 3) ⇔ jump(9, 3) which returns 9+ 3 = 12
jump(**jump(x+3, 3)**, 2) ⇔ jump(**12**, 2) which returns 12 + 2 = 14

6. Write a function isPrime that takes an integer as parameter and returns 1 if the integer is prime and 0 otherwise. A prime number is an integer greater than 1 that does not have any divisors other than 1 and itself. (2 pts)

```
int isPrime(int number) {
    int result = 1;
    for (int x=2; x<number; x++) {
        //If a divisor is found, return 0
        if (number % x == 0)
            return 0;
    }
    //The loop ended without finding any divisors
    return 1;
}
```

Part II: Lab Assignment

Getting started

Change into the cse220 directory
Create a new directory called lab08
Change into the new directory
Implement the program below in your lab08 directory

Sorting Arrays

Part I

Write a program that asks the user to enter a 10 integers and stores them in an array. Write a function (largest) that takes as input an array of int, the length of the array and a position in the array and returns

the largest element of the array between the given position and the end of the array.
Your program should then ask the user for a position and use the function largest to return the largest element starting at that position. Compile and test your program.

Part II

Modify your function, change its name to largestPos and update its code so it returns the position of the largest element found. Adjust your program so it prints the largest element in the array starting at the user given position.

Part III

Write a program (sort.c) that asks the user to enter a number of integers, stores them in an array, and the sorts the array in decreasing order. Your program should do the following:

- Ask the user to enter an integer between 2 and 15. This will represent the number of elements to in the array
- Check if the input is between 2 and 15. If not, your program should keep asking the user to enter an integer between 2 and 15 until a valid input is entered.
- Ask the user to enter the specified number of integers.
- Read those integers into an array
- Print the array before sorting
- Sort the array in decreasing order: (assume that the last index of the array is LAST)
    - Find the largest element of the array (use the function written in part II)
    - Swap the largest element with the element at position 0
    - Find the largest element between position 1 and LAST
    - Swap the largest element with the element at position 1
    - Find the largest element between position 2 and LAST
    - Swap the largest element with the element at position 2
    - Repeat until all elements are sorted
- Print the array after sorting

Compile your program, call the executable sort.
Submit through the handin system the source code sort.c and the executable sort

```c
#include<stdio.h>

int largestPos(int array[], int n, int start);

int main(void) {
	int index, size;

	//Read size, make sure it is beween 2 and 15
	printf("How many integers would you like to sort? (between 2 and 15)\n");
	scanf("%d", &size);
	while (size < 2 || size > 15) {
		printf("The size must be between 2 and 15. Try again:\n");
		scanf("%d", &size);
	}

	//load integers into array from user input
	int array[size];
	printf("Enter %d integers:", size);
	for (index = 0; index < size; index++) {
		scanf("%d", &array[index]);
	}

	//Print before sorting
	printf("Array content before sorting:\n");
	for (index = 0; index < size; index++) {
		printf("%d\t", array[index]);
	}
	printf("\n");

	//Sort
	int sortIdx, pos, temp;
	for (sortIdx = 0; sortIdx < size-1; sortIdx ++) {
		pos = largestPos(array, size, sortIdx);
		temp = array[pos];
		array[pos] = array[sortIdx];
		array[sortIdx] = temp;
	}

	//Print after sorting
	printf("Sorted array content:\n");
	for (index = 0; index < size; index++) {
		printf("%d\t", array[index]);
	}
	printf("\n");
}

int largestPos(int array[], int n, int start) {
	int x;
```

```
    int posMax;
    int max = -9999;
    for (x=start; x<n; x++) {
        if (array[x] > max) {
            max = array[x];
            posMax = x;
        }
    }
    return posMax;
}
```

Handin

The “handin” system has options to allow you to review your files online and to download them. You Should always verify that you submitted the correct files and they were received by the handin system. You can submit files as many times as you like for a particular assignment. Handin will only keep the last version of each file. Remember to submit your files prior to the deadline as you won’t be able to use handin if the deadline has passed.