| CSCE240 Spring 2017: Worksheet 2 | 8 February 2017 |
|---|---|

Write the output of the following code:

**Main.C**

```
#include "prob.h"

int main() {
    int a = 10;
    int b = 10;

    cout << max(a++,
++b) << endl;
    cout << min(a -= 2,
b *= 2) << endl;
}
```

**Prob.C**

```
#include "prob.h"

//Implement Functions
min and max
int min(int a, int b) {
    if (a < b) {
        return a;
    }
    return b;
}

int max(int a, int b) {
    if (a > b) {
        return a;
    }
    return b;
}
```

**Prob.h**

```
#include <iostream>
//Including a System
Library
using namespace std;

int max(int, int);
int min(int, int);
```

```
11
9
```

Write the Output of the Code:

Main.C

```
#include "prob.h"

int main() {
    int a[] = {0,1,2,3,4};
    int a_size = 5;
    int b[] = {5,6,7,8,9};
    int b_size = 5;
    int * c;
    c = sumArray(a, a_size, b, b_size);
    cout << "Array C: ";
    if (c != NULL) {
        for (int i = 0; i < a_size; i++) {
            cout << c[i] << " ";
        }
    }
    cout << endl;
    cout << arraySum(a, a_size) << endl;
    //For the Solutions:
    jumbleArrays(a, a_size, b, b_size);
    cout << "Array A: ";
    for (int i = 0; i < a_size; i++) {
        cout << a[i] << " ";
    }
    cout << endl;
    cout << "Array B: ";
    for (int i = 0; i < b_size; i++) {
        cout << b[i] << " ";
    }
    cout << endl;
    cout << "Largest Number: " << largestNumber(a, a_size) << endl;
    cout << "Smallest Number: " << smallestNumber(a, a_size) << endl;
    cout << "Array Sum of B: " << arraySum(b, b_size) << endl;
}
```

Prob.C

```
#include "prob.h"

//Get the sum of the rows of the array and return an array of the
sums
int* sumArray(const int * a, const int a_size, const int * b, const
int b_size) {
    int * toReturn;
    if (a_size != b_size) {
        return NULL;
    }
    toReturn = new int[a_size];
    for (int i = 0; i < a_size; i++) {
        toReturn[i] = a[i] + b[i];
    }
    return toReturn;
}

//Find summ of all the elements in the array
int arraySum(const int * a, const int a_size) {
    int sum = 0;
    for (int i = 0; i < a_size; i++) {
        sum += a[i];
    }
    return sum;
}
```

Prob.h

```
#include <iostream> //Including a System Library
using namespace std;

//Get the sum of each row of the array and return an array of the
sums
int* sumArray(const int * , const int , const int *, const int );
int arraySum(const int *, const int);

//Write these functions
int largestNumber(const int *, const int);
int smallestNumber(const int *, const int);
//Swaps every other element in the array until it hits the end
void jumbleArrays(int *, const int, int *, const int);
```

Write the Function largestNumber() and smallestNumber() and jumbleArrays().

```
/*  Finds the Largest Number in the Array
 *      const int *a -> Array a
 *      const int a_size -> Size of Array a
 */
int largestNumber(const int *a, const int a_size) {
    int toReturn = a[0];
    for (int i = 0; i < a_size; i++) {
        if (a[i] > toReturn) toReturn = a[i];
    }
    return toReturn;
}

/*  Finds the Smallest Number in the Array
 *      const int *a -> Array a
 *      const int a_size -> Size of Array a
 */
int smallestNumber(const int *a, const int a_size) {
    int toReturn = a[0];
    for (int i = 0; i < a_size; i++) {
        if (a[i] < toReturn) toReturn = a[i];
    }
    return toReturn;
}

/*  Given Arrays A and B, this will swap the elements from every other
index until we hit the end of the smallest array. This will Jumble
Both arrays.
 *      const int *a -> Array a
 *      const int a_size -> Size of Array a
 *      const int *b -> Array b
 *      const int b_size -> Size of Array b
 */
void jumbleArrays(int *a, const int a_size, int *b, const int b_size)
{
    int temp = 0;
    for (int i = 0; i < a_size && i < b_size; i++) {
        if (i % 2 == 1) {
            temp = a[i];
            a[i] = b[i];
            b[i] = temp;
        }
    }
}
```

Output is:

```
Array C: 5 7 9 11 13
10
Array A: 0 6 2 8 4
Array B: 5 1 7 3 9
Largest Number: 8
Smallest Number: 0
Array Sum of B: 25
```