CS 31 Worksheet 4

This worksheet is entirely **optional**, and meant for extra practice. Some problems will be more challenging than others and are designed to have you apply your knowledge beyond the examples presented in lecture, discussion or projects. All exams will be done on paper, so it is in your best interest to practice these problems by hand and not rely on a compiler.

Concepts

Functions, Parameter Passing Arrays

1. Show what will be printed by each of the following programs.

a.
```
#include <iostream.h>
void doglobal();
void dolocal();
void doref(int&);
void doval(int);
int x;
int main()
{
   x = 15;
   doref(x);
   cout << "x = " << x << " after the call to doref\n";
   x = 16;
   doval(x);
   cout << "x = " << x << " after the call to doval\n";
   x = 17;
   dolocal();
   cout << "x = " << x << " after the call to dolocal\n";
   x = 18;
   doglobal();
   cout << "x = " << x << " after the call to doglobal\n";
   return 0;
}
void doref(int& a)
{
   a = 3;
}
void doval(int b)
{
   b = 4;
}
void dolocal()
{
   int x;
```

```
   x = 5;
}
void doglobal()
{
   x = 7;
}
```

b.
```
#include <iostream.h>
int num = 10;
void one();
void two(int);
void three();
void four(int&);
void five(int&);
int main()
{
   int num = 1;
   cout << "At start of main num = " << num << endl;
   one();
   cout << "After call to one num = " << num << endl;
   two(num);
   cout << "After call to two num = " << num << endl;
   three();
   cout << "After call to three num = " << num << endl;
   four(num);
   cout << "After call to four num = " << num << endl;
   two(num);
   cout << "After call to two num = " << num << endl;
   one();
   cout << "After call to one num = " << num << endl;
   five(num);
   cout << "After call to five num = " << num << endl;
   one();
   cout << "After call to one num  " << num << endl;
}
void one()
{
   cout << "   At the start of one num = " << num << endl;
   num = 50;
   cout << "   At the end of one num = " << num << endl;
}
void two(int num)
{
   cout << "   At the start of two num = " << num << endl;
   num = 5;
   cout << "   At the end of two num = " << num << endl;
}
void three()
{
   int num = 100;
   cout << "   At the start of three num = " << num << endl;
```

```
   num = 200;
   cout << "   At the end of three num = " << num << endl;
}
void four(int& num)
{
   cout << "   At the start of four num = " << num << endl;
   num = 25;
   cout << "   At the end of four num = " << num << endl;
}
void five(int& i)
{
   cout << "   At the start of five num = " << num << endl;
   num = 2;
   i = 3;
   cout << "   At the end of five num = " << num << endl;
}
```

c.
```
#include <iostream.h>
void triple(int);
int main(void)
{
   int x;
   for (x = 1; x <= 5; x++)
      triple(x);
}

void triple(int value)
{
   static int total = 0;
   int answer;
   answer = 3 * value;
   total += answer;
   cout << value << ' ' << answer << endl;
   cout << "total " << total
        << endl << endl;
}
```

2. Declare a function named `scan` that reviews an array of int and returns the largest and smallest number found in the array. HINT #1: You'll need to pass an array argument and a companion size parameter. HINT #2: Since you are returning more than one value from your function, you'll need to use reference parameters. Implement this function and then write statements to call this function with an array of size 5.

3. What is the output of the following program?

```
#include <iostream.h>
int main()
{
```

```
   int a[100], b[100], j, m;
   int suma = 0, sumb = 0, sumdiff = 0;
   cin >> m;
   for (j = 0 ; j < m ; j++)
   {
      cin >> a[j] >> b[j];
      suma = suma + a[j];
      sumb += b[j];
      sumdiff = sumdiff + (a[j] - b[j]);
   }
   for (j = m - 1 ; j >= 0 ; j--)
      cout << a[j] << "  " << b[j] << "  " << a[j] - b[j] << endl;
   cout << suma << "  " << sumb << "  " << sumdiff << endl;
}
DATA:
5
11 15
19 14
4 2
17 6
1 3
```

4. Given:
```
int h = 6, p = 2, m = 3;
int values[7];
```
Suppose values contains: -4 0 2 6 -2 -1 14
Show the contents of the array values after:
```
for (; m <=5; m++)
       values[m] = values[h] + values[p] * values[m];
```

5. Given the declarations:
```
int sample[8], i, k;
```
show the contents of the array sample after the following code is executed. Use a question mark to indicate any garbage values in the array.

```
for (k = 0 ; k < 8 ; k++)
   if (k % 2)
      sample[k] = 1;
```

6. What is the error in the following program segment?

```
int main()
{
   int i, count[10];
   cout << "please enter 10 numbers:  ";
   for (i = 1; i <= 10; i++)
      cin >> count[i];
}
```

7. Write the statements to multiply every element of an array of ints (of size 50) by 2.

8. Write the statements to add up those elements of an array of ints (of size 25) which have an even subscript.

9. Write the statements to add up those elements of an array of ints (of size 25) which have an even value.

10. What will the following program segment print?

```
int main()
{
   int nums[10];
   int i;
   for (i = 9 ; i >= 0 ; i--)
   {
      nums[i] = 5 * (i + 1);
      cout << nums[i] << "  ";
   }
   cout << endl;
   for (i = 0 ; i < 9 ; i++)
      cout << nums[i] << "  ";
   cout << endl;
   for (i = 0 ; i < 9 ; i++)
      nums[i+1] = nums[i];
   for (i = 0 ; i < 9 ; i++)
      cout << nums[i] << "   ";
   cout << endl;
}
```

11. What will the following program segment print?

```
int main()
{
   int nums[10];
   int i;
   for (i = 9 ; i >= 0 ; i--)
   {
      nums[i] = 5 * (i + 1);
      cout << nums[i] << "  ";
   }
   cout << endl;
   for (i = 0 ; i < 9 ; i++)
```

```
        cout << nums[i] << "  ";
    cout << endl;
    for (i = 8 ; i >= 0 ; i--)
        nums[i+1] = nums[i];
    for (i = 0 ; i < 9 ; i++)
        cout << nums[i] << "   ";
    cout << endl;
}
```

12. Given: `int temps[50];`
Write the statements to print "yes" if any element of the array temps contains the value 100.

13. Given: `int temps[50];`
Write the statements to set the variable found to true if any element of the array temps contains the value 100. If not, the variable found should be false.