Name: _____

Write all of your responses on these exam pages. If you need extra space please use the backs of the pages.

1 Short Answer: 3 Points Each

1. Write a function prototype for a function named `fct` that returns a string and takes in parameters of types of one int and two doubles in that order. The doubles are to be passed by reference and the integers by value.

2. Write a function prototype for a function named `fct` that returns nothing and takes in a single long integer parameter that is defaulted to 1000.

3. Given the following declaration,

   ```
   int A[10] = {2, 3, 5, 7, 11, 13};
   ```

   What are the values of

   (a) `A[3]`

   (b) `A[0]`

   (c) `A[9]`

4. Given the following function,

```
int justdoit(int A[], int size) {
    static int s = 0;
    for (int i = 0; i < size; i++)
        s += A[i];
    return s;
}
```

and the following code from the main,

```
int A[] = { 5, 7, 9 };
int B[] = { 1, 2, 3, 4 };
int C[] = { 10, 9, 8, 7, 6, 5 };
justdoit(A, 3);
justdoit(B, 4);
cout << justdoit(C, 4) << endl;
```

What is the output?

5. If you have a global variable named x that has value 7 and a local variable in a function (not in the main) also named x that has value 5 and you print out the value of x from the function what will the output be?

6. Write a for-each loop (a.k.a. ranged-based loop) that will populate an integer array A (previously defined) with random numbers between 1 and 100. Also write another for-each loop (a.k.a. ranged-based loop) that will print out this array.

7. If we define an array as `int A[5] { 5, 7, 9 };` what would be the output of `cout << A + 2 << endl;`

8. When writing a function that accepts a two-dimensional array as an argument, which size declarator must you provide in the parameter for the array?

9. Write a single line declaration of a two-dimensional integer array that has 2 rows and 4 columns and is initialized with the first row containing 1, 2, 3, 4, and the second row containing 9, 8, 7, 6.

10. Say we have the following function,

```
int justdoit(int a, int b) {
    return a + b;
}
```

and we have the following line in the main,

```
cout << justdoit(5, justdoit(2, 6)) << endl;
```

Is this syntactically correct and if so what is the output?

2 Program Traces: 10 Points Each

1. Write the output of the following program.

```
#include <iostream>
#include <string>
using namespace std;

int doit(int a, int &b) {
    cout << "intint" << endl;
    cout << a << " " << b << endl;
    for (int i = 0; i < b; i++)
        a += a;
    b -= 5;
    return a + b;
}
string doit(string a, int b) {
    cout << "stringint" << endl;
    cout << a << " " << b << endl;
    for (int i = 0; i < b; i++)
        a += a;
    return a;
}
int doit(int a, char &b) {
    cout << "intchar" << endl;
    cout << a << " " << b << endl;
    a += b - 'A';
    b += 3;
    return a;
}
int main() {
    int a = 3;
    int b = 2;
    string s = "Hi";
    char c = 'F';
    cout << a << " " << b << endl;
    cout << doit(a, b) << endl;
    cout << a << " " << b << endl;
    b = 2;
    cout << "----------" << endl;
    cout << s << " " << b << endl;
    cout << doit(s, b) << endl;
    cout << s << " " << b << endl;
    cout << "----------" << endl;
    a = 4;
    cout << a << " " << c << endl;
    cout << doit(a, c) << endl;
    cout << a << " " << c << endl;
    cout << "----------" << endl;
    cout << a << " " << b << endl;
    cout << c << " " << s << endl;
    cout << doit(doit(a, c), b) << endl
        ;
    cout << a << " " << b << endl;
    cout << c << " " << s << endl;
    return 0;
}
```

Output

2. Write the output of the following program.

```
#include <iostream>
using namespace std;

void swap(int &a, int &b) {
    int temp = a;
    a = b;
    b = temp;
}

int main() {
    int size = 7;
    int A[size] = { 5, 4, 3, 2, 1 };

    for (int i = 0; i < size; i++)
        cout << A[i] << " ";
    cout << endl;

    for (int i = 0; i < size - 1; i++)
        A[i + 1] += A[i];

    for (int i = 0; i < size; i++)
        cout << A[i] << " ";
    cout << endl;

    for (int k = 0; k < 3; k++) {
        int temp = A[0];
        for (int i=0;i<size-1;i++)
            A[i] = A[i + 1];
        A[size - 1] = temp;
    }

    for (int i = 0; i < size; i++)
        cout << A[i] << " ";
    cout << endl;

    cout << A[size / 2] << endl;

    for (int i = 0; i < size; i++)
        A[i] = size - i;

    for (int i = 0; i < size; i++)
        cout << A[i] << " ";
    cout << endl;

    for (int i = 0; i < 3; i++) {
        int a = (3 * i) % size;
        int b = (3 * i + 1) % size;
        swap(A[a], A[b]);
    }

    for (int i = 0; i < size; i++)
        cout << A[i] << " ";
    cout << endl;

    return 0;
}
```

Output

3 Programming: 10 Points Each

For each of these, just write the function, you may assume that the main declares everything that is needed on its end and that the needed includes have been made.

1. Write a function that takes as input an array of doubles and the size of the array and returns the largest value in the array. Make sure that the array is read only in the function.

2. Write a function that takes as input an array of integers and the size of the array and reverses the array. That is, the array entries are in reverse order when the function ends. Do this without creating a second array inside the function.

3. Write a function called `getInteger` that takes three parameters, a string that will be used as a prompt for user input, and two integers that are the lower bound and upper bound for the input. When called, the function will prompt the user with the parameter prompt, take an integer input, and determine if it is between the lower and upper bounds (inclusively). If it is the function will return the input, otherwise it will print an error that includes an instruction to the user on the range that is acceptable, and get input from the user again, until a legitimate value is input. For example, if the main has the following call,

```
int t = getInteger("Input a number: ", 5, 10);
cout << t << endl;
```

The user interface would be look like the following.

```
Input a number: 12
Value out of range, must be in 5 to 10.
Input a number: -4
Value out of range, must be in 5 to 10.
Input a number: 17
Value out of range, must be in 5 to 10.
Input a number: 10
10
```

4. Write a function called `wordcount` that takes in two parameters, a string that represents a filename and another string that is a single word target. The function will open the file, which is just a text file, read in the file word for word and count the number of words that match the target word. This count will be returned by the function. The matching is to be case insensitive and all punctuation is to be removed. So if a string like `her's,` is read from the file it would be converted to `HERS` before tested for a match.

5. Write a function that takes as its only input parameter a string that is the filename of a data file. The data file consists of a 2-D grid of numbers. The number of rows and the number of columns varies from file to file. So one file may look like,

```
795 570 434 378 467
601 97 902 317 492
652 756 301 280 286
441 865 689 444 619
```

and another like,

```
270 699 417 839 569 363 622 794 173 847
431 462 682 390 292 791 57 115 521 157
574 491 947 951 231 21 537 740 54 30
98 325 81 516 516 2 231 139 796 404
338 580 218 21 970 862 812 379 977 685
536 904 176 483 207 759 857 744 499 911
127 950 236 560 818 105 563 49 244 711
805 934 291 375 955 614 589 768 993 918
805 882 822 982 717 30 93 574 126 593
486 253 543 74 814 713 179 377 762 775
88 919 710 732 294 17 346 235 137 691
153 943 573 328 925 291 710 18 217 836
963 55 90 858 130 904 571 661 633 685
```

The entries in the file are separated by tabs and there is a tab after each entry. Each row has the same number of items in it. Have your program open the data file and read the data into a two-dimensional array that has the same number of rows and columns as the data layout in the file.

Once the data is in the array create another array, one-dimensional, whose size is the same as the number of rows of the data array you created, this will be your sums array. For each row of the data array find the row sum, sum of all the entries in the row, and place that in the corresponding position of the one-dimensional array. That is, the i^{th} entry of the sums array will contain the sum of the entries in the i^{th} row of the data array, this is a parallel array setup. So for the second example above the sums array would look like,

```
5593 3898 4576 3108 5842 6076 4363 7242 5624 4976 4169 4994 5550
```

Finally, write the sums array entries, one per line, to a file names `sums.txt`.