Assignment 4 01/11/2-07

1. What does the following program print out? (Answer without using the computer)

```
void f( const int a = 5 )
{
  std::cout << a * 2 << endl;
}

int a = 123;
int main()
{
  f(1);
  f(a);
  int b = 3;
  f(b);
  int a = 4;
  f(a);
  f();
}
```

2. Identify the error(s) and explain how you would correct them to do what it is apparently meant to do. Give two ways to fix this one.

```
#include <iostream>
int main() {
  printNum(35);
  return 0;
}

void printNum(int number) { std::cout << number; }
```

3. Identify the error(s) and explain how you would correct them to do what it is apparently meant to do. Give two ways to fix this one, and explain which is preferable.

```
#include <iostream>

void printNum() { std::cout << number; }

int main() {
  int number = 35;
  printNum(number);
  return 0;
}
```

4. Identify the error(s) and explain how you would correct them to do what it is apparently meant to do.

```
#include <iostream>

void doubleNumber( int num ) { num = num * 2; }

int main() {
  int num = 35;
  doubleNumber(num);
  std::cout << num;  // Should print 70
  return 0;
}
```

5. Identify the error(s) and explain how you would correct them to do what it is apparently meant to do.

```
#include <iostream>
#include <cstdlib>  // Contains some math functions

int difference( const int x, const int y ) {
  int diff = abs(x - y);  // abs(n) returns absolute value of n
}
int main() {
  std::cout << difference(24, 1238);
  return 0;
}
```

6. Write the following C++ global functions:

   a. Write a function sum that returns the sum of two integers also write the equivalent function for taking the sum of two doubles.
   b. Explain why, given your functions from (a), sum(1, 10.0) is a syntax error.
   c. Write just one function that, using default arguments, allows you to take the sum of anywhere between 2 and 4 integers. What would happen if you put both this definition and a 3 argument function (that computes the sum of three integers) into the same file, and called sum(3, 5, 7)? Why?
   d. Write a single sum function capable of handling and arbitrary number of integers using an array.
   e. Rewrite (d) to use recursion instead of a loop.

7. Using a "Monte Carlo" method – that is, a randomized simulation – we can compute a good approximation of π. Consider a circle of radius 1, centered on the origin and circumscribed by a square. Imagine that this is a dartboard and that you are tossing darts at it randomly. With enough darts, the ratio of darts in the circle to total darts thrown should be the ratio between the area of the circle (call it a) and the area of the square (4):$\frac{total\ darts}{dart\ in\ circule} = \frac{4}{a}$
   We can the use this ratio to calculate a, from which we can then find: $\pi = \frac{a}{r^2}$
   We can simplify the math by only considering the first quadrant, calculating the ratio of the top right quadrant of the square to the area of one quarter of the circle. Thus, we will actually find $\frac{a}{4}$ and compute:$\pi = 4 \times \frac{a/4}{r^2}$
   We'll build a function to do this. Define a function that takes a parameter n which has a default value of 1000000, and returns a double. Use a loop and a random number generator to simulate n random "throws" of the dart, computing the sum of the "hits" using an unsigned long int. You can determine whether a dart hits the circle using the Euclidean distance formula: . Test your function using different values of n and determine how close you can get to an accurate value for .

8. Write a function printArray to print the contents of an integer array with the string ", " between elements, but not after the laste element. your function should return nothing.
9. Write a reverse function that takes an integer array and its length as arguments. Your function should reverse the contents of the array, leaving the reversed values in the original array, and return nothing.

For all problems, put your solution in a text file. Submit online a single file that contains all functions for problems 6-9. The program must run and test each function.