QUEEN MARY, UNIVERSITY OF LONDON

MTH6150

Numerical Computing in C and C++

Exercise Sheet 5

1. Try running the code on slide 8 of lecture 5, and check what error message you see when compiling it. Also, what happens if you declare a `const` variable without giving it a value?

2. Type in the function `triangle_area` on slide 14 of lecture 5.

   Check that the function produces the expected output with various sets of three numbers, as on slide 13.

   Add code to output an error message to the screen saying that this is not a valid triangle if `area2<0` - this command needs to be immediately before `return -1;`

   Modify the code so that the function `triangle_area` is declared before the `main` function, but the `triangle_area` definition is after the `main` function.

3. Write a function to calculate the hypotenuse `z` of a right-angled triangle

   $$z = \sqrt{x^2 + y^2}$$

   ```
   double my_hypot(const double x, const double y){
       ...
   }
   ```

   Again, check that the results are correct.

4. Type in the code for each of the functions on slide 24, lecture 5, and check that the results are what you expect.

5. Use the function `is_even` from the preceding question in a program that does the following:

   - declares an `int` variable;
   - outputs a message asking the user to input an odd number;
   - and then if the input number is even, repeatedly outputs a message saying that the input needs to be odd, until an odd number is entered (see slide 25).

   Put this inside a function

   ```
   int get_odd_int(){
       ...
   }
   ```

   that gets an odd integer from the user of the program.

6. Write an alternative to the function on slide 27

   ```
   double get_pi(){
       ...
   }
   ```

   which uses the fact that

   $$\mathit{tan}\,(\pi/4) = 1$$

   and hence

   $$\pi = 4\ \mathit{arctan}(1)$$

   Note that *arctan* is available in C++ as `atan` when you have

   ```
   #include <cmath>
   ```

7. Write a function `sum` that takes an integer argument n and returns the sum of the integers from 1 to n. You can check your result against the formula

   $$\sum_{k=1}^{n} k = \frac{1}{2}n(n+1)$$

   Hint: see exercise sheet 4, question 1b. `S` would be returned from the function instead of output to the screen.

8. Write a function that returns the factorial $n!$ using a `for` loop inside the function.

   ```
   int factorial(const int n){
       ...
   }
   ```

   Make sure that it allows the value $n = 0$ as an argument, which should return the value $0! = 1$, as well as allowing positive integer values of n.

   Hint: You could start with an `int` variable equal to 1 (before the loop); then use the `for` loop to multiply the variable by $2, 3, \ldots, n$; and then return the result.

9. Write a function of the following form

   ```
   bool is_prime(const int n){
       ...
   }
   ```

   which returns `true` if `n` is prime and `false` otherwise.

   Test this function using code such as the following:

   ```
   for(int n=1; n<=20; n++){
       if(is_prime(n)){
           cout << n << endl;
       }
   }
   ```