cmath q2

# Revision history

V 1.0: initial release

# usage history

# problem description

Predict the output of the following code, then test the code and see if your output was correct. If not, what went wrong?

#include <iostream>  
#include <cmath>

**using** **namespace** std**;**

int main**()**

**{**

cout **<<** cos**(**0**)** **<<** endl**;**

cout **<<** sin**(**M\_PI**/**2**)** **<<** endl**;**

cout **<<** cos**(**90**)** **<<** endl**;**

cout **<<** 1**/**2**\***sin**(**M\_PI**/**2**)** **<<** endl**;**

cout **<<** pow**(**10**,**1**/**2**)** **<<** endl**;**

cout **<<** sqrt**(**225**)** **<<** endl**;**

cout **<<** sqrt**(-**1**)** **<<** endl**;**

**}**

# Solution

* The first line should output 1, since 0 rad = 0 degrees.
* The second line should output 1, since the argument to sin is in radians and I know that sin(M\_PI/2 ) = 1 from high school math.
* The third line will output -0.45 or something close. If you predicted that you’d see a zero, remember that the argument to sin and cos must be in radians, so the input to the function was 90 rad, not 90 degrees.
* The fourth line will output zero, since the ½ term is evaluated using integers 1 and 2, and therefore this suffers from an integer division error.
* The fifth line will output 1 for the same reason. The ½ term is evaluated using integers 1 and 2, which evaluates to 0. Then the exponent is evaluated. Anything to the exponent 0 is 1.
* The sixth line will output 15 (or a suitably close rounding of that number, depending on student computer).
* The seventh line will output “nan”, or “not a number”. This is also a silent failure, since the computer did not obviously indicate that the input was bad.

# suggested test cases

None.

# required topics

* Floating point rounding
* Integer division
* Use of pow, sqrt, cos, sin
* Use of M\_PI constant
* Radians are the input to trig functions