```
// #1
template <typename T>
T cube(T value)
{
  return value * value * value;
}

// #2
template <>
int cube<int>(int value)
{
  return value * value * value;
}

// #3
int cube(int value)
{
  return value * value * value;
}
```

1. Given the three functions above, which function will be called by each statement below. Indicate the function by writing the number (**1**, **2**, or **3**) next to the call. If the call is invalid, write **INV**.

   a) _____ `cube<double>(10L);`

   b) _____ `cube<int>(2.5);`

   c) _____ `cube(2.5F);`

   d) _____ `cube(5);`

   e) _____ `cube<char>(5);`

2. What is the output of the program below?

```
void DivideByZero()
{
  cout << "Line 6" << endl;
  int a = 5, z = 0;
  if (z == 0)
    throw "Divide by 0";
  int b = a / z;
  cout << "Line 7" << endl;
}
```

```
int main()
{
  cout << "Line 1" << endl;
  try
  {
    cout << "Line 2" << endl;
    DivideByZero();
    cout << "Line 4" << endl;
  }
  catch (const char *message)
  {
    cout << message << endl;
  }
  cout << "Line 5" << endl;
  return 0;
}
```

More →

3. For each of the following, if the problem can be handled with C++ exceptions write **YES.** If it can't be handled by an exception, write **NO.**

   a) _____ Attempts to divide by zero

   b) _____ A missing semi-colon at the end of a statement

   c) _____ Taking the square root of –345.26

   d) _____ Misspelled C++ keywords

   e) _____ An invalid array subscript in a user-defined object

```
class A                          class B : public A
{                                {
  public:                          public:
    A(int x = 0) { a_ = x; }         B(int x) { a_ = x; }
    void f1()                        void f1(int)
    {                                {
      std::cout << "A1";               std::cout << "B1";
    }                                }
    void f2()                        void f3()
    {                                {
      std::cout << "A2";               std::cout << "B3";
    }                                }
    void f3(int)                     void f4()
    {                                {
      std::cout << "A3";               std::cout << "B4";
    }                                }
  private:                         private:
    int a_;                          int a_;
};                               };
```

4. Given the classes above, what is the output from the statements below? If the statement does not compile, write **NC.**

```
A a;
B b(5);

a.f1();   a) _____
b.f1();   b) _____
b.f2();   c) _____
a.f3();   d) _____
b.f1(5);  e) _____
```