**Problem 1 [15 points]:**

Compare the exception domains of the following apparently identical pieces of code (the variable Initial is of type integer):

**procedure Do_Something is**

- **subtype Small_Int is Integer range -16..15;**
- A; Small_Int := Initial;

**begin**

...  
**end Do_Something;**

**procedure Do_Something2 is**

- **subtype Small_Int is Integer range -16..15;**
- A; Small_Int;

**begin**

A := Initial;

...  
**end Do_Something2;**

**procedure Do_Something3 is**

- **subtype Small_Int is Integer range -16..15;**
- A; Small_Int := Initial;

**begin**

begin

A := Initial;

...  
end;

**end Do_Something3;**


**Problem 2 [15 points]:**

In a process control application, gas is heated in an enclosed chamber. The chamber is surrounded by a coolant which reduces the temperature of the gas by conduction. There is also a valve which when open releases the gas into the atmosphere. The operation of the process is controlled by an Ada package whose specification is given below. For safety reasons, the package recognizes several error conditions; these are brought to the notice of the user of the package by the raising of exceptions. The exception Heather_Stuck_On is raised by the procedure Heater_Off when it is unable to turn the heater off. Similarly other two exceptions are listed below.

**package Temperature_Control is**

- **Heater_Stuck_On, Temperature_Still_Rising, Valve_Stuck: exception;**
procedure Heater_On;
    -- turn on heater
procedure Heater_Off;
    -- turn off heater
    -- raises Heater_Stuck_On
procedure Increase_Coolant;
    -- causes the flow of coolant which surrounds the chamber to increase until
temperature reaches a safe level
    -- raises Temperature_Still_Rising alarm
procedure Open_Valve;
    -- opens a valve to release some of the gas there by avoiding an explosion
    -- raises Valve_Stuck
procedure Panic;
    -- sounds an alarm and calls the fire services
end Temperature_Control;

Write an Ada (or C, C++ if you like) procedure which when called will attempt to turn off
the heater in the gas chamber. If the heater is stuck on then the flow of coolant surrounding
the chamber should be increased. If the temperature is still rising then the escape valve
should be opened. If all fails, then the alarm must be sounded.

Problem 3 [10, 10 points]: Time and Value Domain Errors
Are timing and value failures orthogonal? You can answer this question by explaining the
following two questions: (a) explain how a system can be transformed so that all timing
failures manifest themselves as value failures; (b) explain if the converse of (a) can be
achieved? Is this universally true, or explain the circumstances under which the converse is
ture.

Problem 4 [10 points]: Providing delay through blocking
Using your favorite language, implement a monitor interface to a simple (coarse) delay
mechanism consisting to two procedures tick() and delay(). A caller to delay(int d) wishes
to be suspended for d ticks. The procedure tick() is called by some clock routine. Whenever
it is called each process blocked on delay() wakes up, decrements some counter (which is
initialized to d) and exits the monitor if the counter has reached zero – otherwise it
reblocks.

Show how the body of the monitor can be implemented.

Hint: You can use condition variables that have wait and signal operations defined on
them.