Concurrent Programming **Museum**
Estimated Time: 1.0 hrs **By: October 2015**

A museum, which can hold a maximum of **N** visitors, allows visitors to enter through the east entrance and leave through its west exit. Arrivals and departures are signalled to the museum controller by the turnstiles at the entrance and exit. At opening time, the museum director signals the controller that the museum is open and then the controller permits both arrivals and departures. At closing time, the director signals that the museum is closed, at which point only departures are permitted by the controller. When all visitors have left the museum, the controller signals the director that the museum is empty. The director cannot reopen the museum unless it is empty.

The alphabet of actions for the museum is:

`enter`	- visitor enters through the east entrance
`exit`	- visitor leaves through the west exit
`open`	- the director opens the museum
`close`	- the director closes the museum
`empty`	- the controller signals that the museum is empty

The processes required to model the museum system are:

`EAST`	- museum entrance
`WEST`	- museum exit
`CONTROL`	- museum controller
`DIRECTOR`	- museum director

To do:

1. Draw a structure diagram which shows the relationship between the processes and actions in the museum system. <*2 marks*>

2. Write a specification in FSP which models the behaviour of the museum system i.e. provide a definition for each of the processes and a definition of how they are composed to form the system. <*5 marks*>

3. Specify a `TEST` process, which when composed with the museum system, checks that it is not possible for visitors to enter when the museum is closed.
(hint: use `ERROR`). <*2 marks*>

4. List the trace which you should obtain when the following process is composed with the museum system and the **Safety** menu in LTSA is used:
```
P = STOP + {exit}.
```
<*1 mark*>
(Note: the process P prevents the action exit from ever happening.)