

SOLUTION

A clinic has **T treatment rooms** and **D doctors**, each serving one patient at a time. Patients are classified as either **Normal** or **Emergency**.

- A **Normal patient** requires only **one** doctor, whereas an **Emergency patient** requires **two (2)** doctors.
- A patient can be served only if both a treatment room and the required number of doctors (one for Normal, two for Emergency) are available; otherwise, they must **wait**.
- **Emergency patients have priority** over Normal patients and must be served first. Specifically, a Normal patient must wait if there is at least one waiting Emergency patient, even if a doctor is currently available.

Normal and **Emergency** patients can be considered parallel processes, while **treatment rooms** and **doctors** are shared resources. Complete the monitor-based synchronization algorithm below for this problem.

<pre>// specify the procedures in the monitor CLINIC Process Normal-Patient (i) begin CLINIC.NormalEnter() ; RECEIVE-TREATMENT (); CLINIC.NormalExit() ; end;</pre>	<pre>// specify the procedures in the monitor CLINIC Process Emergency-Patient (j) begin CLINIC.EmergencyEnter() ; RECEIVE-TREATMENT (); CLINIC.EmergencyExit() ; end;</pre>
<pre>//give the monitor variables declaration Monitor CLINIC VAR freeRooms: integer; freeDoctors: integer; countE:integer ; E: condition; N: condition;</pre>	<pre>// give the monitor variables initialization Begin freeRooms = T; freeDoctors = D; countE:=0; End;</pre>
<pre>// declare the different procedures of your monitor (you can also use the back of the sheet to write your procedures) procedure EmergencyEnter() begin countE + +; while (freeRooms =0 OR freeDoctors < 2) E.wait(); freeRooms - - ; freeDoctors = freeDoctors – 2; countE - - ; end;</pre>	

```
procedure EmergencyExit()
begin
    freeRooms + + ;
    freeDoctors = freeDoctors + 2;
    E.signal;
    N.signal;
end ;

procedure NormalEnter()
begin
    while (freeRooms =0 OR freeDoctors =0 OR countE>0) N.wait;
    freeRooms- -;
    freeDoctors - -;
end;

procedure NormalExit()
begin
    freeRooms + +;
    freeDoctors + +;
    E.signal;
    N.signal;
end;
```