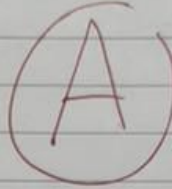


1) Sem mutexCola = 1;
 Sem mutexTropical = 1;
 sem mutexNoTropical = 1;
 Cola colaData [365];
 Cola colaTropical;
 Cola colaNoTropical;



1	B
2	R+
3	B

Process Worker [id: 0..4]{

 text data;

 P(mutexCola);

 while (not colaData.isEmpty()){

 c.pop(data);

 V(mutexCola);

 boolean tropical = Process(data);

 if (tropical){

 P(mutexTropical);

 colaTropical.push(data);

 V(mutexTropical);

 } else{

 P(mutexNoTropical);

 colaNoTropical.push(data);

 V(mutexNoTropical);

 }

 P(mutexCola);

}

V(mutexCola);

}

```

2) sem mutexPedido = 1;
   sem HayPedido = 0;
   sem espera[N] = ([N], 0);
   cola c;
   boolean pide[N];

```

```

Process Cliente [i: 0..N-1] {
    while (true) {
        P(mutexPedido);
        c.push(i);
        V(mutexPedido);
        V(hayPedido);
        P(espera[i]);
        Writein (pide[i]);
    }
}

```

```

Process Impresora {
    while (true) {
        P(hayPedido);
        P(mutexPedido);
        c.pop(i);
        V(mutexPedido);
        text Pedido = ObtenerPedido(i);
        boolean OK = verificarPieza(Pedido);
        pide[i] = OK;
        V(espera[i]);
    }
}

```

hacer el servicio

debe cliente 7 le cote

hacer el check al recibir la pieza

debe imprimir la pieza

3) Process Persona [id: 0..49] {

 int eoad = obtenerEoad();

 AdminCabina.llegar(id, eoad);

 tomarfoto();

 AdminCabina.salir();

}

Monitor AdminCabina {

 cond espera[50]; int id;

 boolean libre = true;

 int esperando = 0;

 colaOrdenada c;

 Procedur2 llegar(id: in int, eoad: in int) {

 if (not libre) {

 esperando++;

 c.pushordenado(id, eoad); // Agregar a pers. mayor eoad

 wait(espera[id]);

 }

 } else {

 libre = false;

 }

}

 Procedur2 salir() {

 if (esperando > 0) {

 esperando--;

 c.pop(id);

 Signal(espera[id]);

 } else {

 libre = true;

 }

}