

## Second set of sample exam questions

**Gabriel Moise**

1.

(a)

```
void rand_task (unsigned arg) {  
    message m;  
  
    int stack[MAX];  
  
    int stack_length = 0;  
  
    // enable interrupts from hardware and connect them  
    while(1) {  
        if (stack_length == 0) receive(HARDWARE, &m);  
        else receive(ANY,&m);  
  
        switch(m.m_type) {  
            case INTERRUPT : {  
                if (! RNG_VALRDY) break;  
  
                // clear the pending state + re-enable  
                if (stack_length < MAX)  
                {  
                    stack[stack_length] = RNG_VALUE;  
  
                    stack_length++;  
                }  
  
                clear_pending(RNG_IRQ);  
  
                enable_irq(RNG_IRQ);  
  
                RNG_VALRDY = 0;  
            }  
  
            case REQUEST : {  
                message new_m;  
  
                new_m.m_type = REPLY;  
  
                new_m.m_i1 = stack[stack_length-1];  
  
                stack_length --;  
            }  
        }  
    }  
}
```

```

        send(m.m_sender,&new_m);
    }
}
}
}

```

```

unsigned randbyte(void) {
    message m;
    m.m_type = REQUEST;
    sendrec(RANDOM,&m);
    return m.m_i1;
}

```

```

void user_task (unsigned arg) {
    unsigned result = randbyte()
    serial_printf("Random number: %d",m.m_i1);
}

```

```

void init(void) {
    serial_init()
    start(RANDOM,"Random",rand_task,0,STACK)
    start(USER,"User",user_task,0,STACK)
}

```

(b) Calling randbyte several times might be too much as the speed of producing random bytes by the hardware might be too slow for the high demand and the program might need to wait too much, and in the meantime a new command might come and it might be lost.

Therefore, I changed the new\_m message to also fill in the i2,i3,i4 fields.

```

case REQUEST : {
    message new_m;
    new_m.m_type = REPLY;
    new_m.m_i1 = stack[stack_length-1];
}

```

```
stack_length --;  
new_m.m_i2 = stack[stack_length-1];  
stack_length --;  
new_m.m_i3 = stack[stack_length-1];  
stack_length --;  
new_m.m_i4 = stack[stack_length-1];  
stack_length --;  
send(m.m_sender,&new_m);  
}
```