```
#Lab 5 (Thu) solutions
#Suprakash Datta

#Q1:
myMedian :=proc(a :: integer, b :: integer, c :: integer)
 if a ≤ b ≤ c  then return b; #enumerate all possible orders of 3 elements
 elif  a ≤ c ≤ b  then return c;
 elif b ≤ c ≤ a then return c;
 elif b ≤ a ≤ c then return a;
 elif c ≤ a ≤ b then return a;
 elif c ≤ b ≤ a then return b;
 else print("Error");
 end if;
end proc;
```

```
proc(a::integer, b::integer, c::integer)                                    (1)
    if a <= b and b <= c then
        return b
    elif a <= c and c <= b then
        return c
    elif b <= c and c <= a then
        return c
    elif b <= a and a <= c then
        return a
    elif c <= a and a <= b then
        return a
    elif c <= b and b <= a then
        return b
    else
        print("Error")
    end if
end proc
```

```
#Q2

numdig := n→1 + trunc(log10(n));#get the number of digits of an integer
                                  n→1 + trunc(log10(n))                     (2)
numdig(102);
                                  3                                         (3)
ithdig :=proc(n :: integer, i :: integer)#get the ith digit(from the right) of an integer
 return iquo(irem(n, 10^i), 10^(i − 1));
 end proc;
        proc(n::integer, i::integer) return iquo(irem(n, 10^i), 10^(i − 1)) end proc   (4)
 digset :=proc(n :: integer)
local m;
 m := abs(n);#takes care of negative integers
```

```
{seq(ithdig(m, i), i = 1 ..numdig(m))};
 end proc;
    proc(n::integer) local m; m := abs(n); {seq(ithdig(m, i), i = 1 ..numdig(m))} end proc                    (5)
digset(-434); digset(433);
                                          {3, 4}
                                          {3, 4}                                                             (6)
#Q3 Note the extra check if all digits are used. You will not be penalized if you did not do this.
q4 := n→piecewise({0, 1, 2, 3, 4, 5, 6, 7, 8, 9}\ digset(n) ≠ { }, max({0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
     \ digset(n)), {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}\ digset(n) = { }, "All digits used");
n→piecewise({0, 1, 2, 3, 4, 5, 6, 7, 8, 9}\digset(n) ≠ { }, max({0, 1, 2, 3, 4, 5, 6, 7, 8, 9}           (7)
     \digset(n)), {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}\digset(n) = { }, "All digits used")
q4(23456789); q4(1234567890);
                                              1
                                          "All digits used"                                                  (8)
#Q4
NotPowerOf3 :=proc(n :: integer)
return {seq(i, i = 1 ..n)} minus {seq(3^i, i = 0 ..n)};
    #could have restricted the range to 1 +trunc(log3(n))
end proc;
      proc(n::integer) return minus({seq(i, i = 1 ..n)}, {seq(3^i, i = 0 ..n)}) end proc                     (9)
NotPowerOf3(20);
                          {2, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}                     (10)


#Q5 -- make sure you return 1 even if all the digits are 1 or 2
All12 := n→piecewise(digset(n) minus {1, 2} = { }, 1, digset(n) minus {1, 2} ≠ { }, 0);
            n→piecewise(digset(n)\{1, 2} = { }, 1, digset(n)\{1, 2} ≠ { }, 0)                                (11)
All12(21112);
                                              1                                                              (12)
All12(2111123);
                                              0                                                              (13)
All12(222);
                                              1                                                              (14)
All12(111);
                                              1                                                              (15)
```