```
> #Lab 6 Tue solutions
>
> #Q1
> seleven :=proc(L :: list)
  local num, LL;
  LL := [seq(0, i = 1 .. nops(L)/2)];
  for num from 2 to nops(L) by 2 do
  LL[num/2] := L[num];
  end do;
  return LL;
  end proc;
```

```
seleven := proc(L::list)                                                (1)
    local num, LL;
    LL := [seq(0, i = 1 ..1 / 2 * nops(L) )];
    for num from 2 by 2 to nops(L) do LL[1 / 2 * num] := L[num] end do;
    return LL
end proc
```

```
> Lodd := [3, 5, 7, 9, 11, 13, 16];
  Leven := [3, 4, 5, 6, 7, 8];
```

```
                    Lodd := [3, 5, 7, 9, 11, 13, 16]
                   Leven := [3, 4, 5, 6, 7, 8]                          (2)
```

```
> seleven(Lodd);
```

```
                          [5, 9, 13]                                    (3)
```

```
> seleven(Leven);
```

```
                          [4, 6, 8]                                     (4)
```

```
>
> #Q2
> q2 :=proc(L :: list)
  local num, LL;
  LL := [ ];
  for num from 2 to nops(L) do
  if L[num] > 2·L[num − 1] then LL := [op(LL), num];end if;
  end do;
  end proc;
```

```
q2 := proc(L::list)                                                     (5)
    local num, LL;
    LL := [ ];
    for num from 2 to nops(L) do
        if 2 * L[num − 1] < L[num] then LL := [op(LL), num] end if
    end do
end proc
```

```
> L := [1, 2, 5, 6, 7, 15];
```

```
                      L := [1, 2, 5, 6, 7, 15]                          (6)
```

```
> q2(L);
```

$$[3, 6] \quad (7)$$

```
>
> #Q3
> q3 :=proc(L :: list)
  local num, LL;
  LL := [ ];
  if irem(nops(L), 2) ≠ 0 then print("Error: odd length list"); return LL;
  else
   for num from 1 to nops(L) by 2 do
    LL := [op(LL), L[num]];
   end do;
  for num from 2 to nops(L) by 2 do
    LL := [op(LL), L[num]];
   end do;
  return LL;
  end if;
   end proc;
q3 := proc(L::list)                                                    (8)
  local num, LL;
  LL := [ ];
  if irem(nops(L), 2) <>0 then
      print("Error: odd length list"); return LL
  else
      for num by 2 to nops(L) do LL := [op(LL), L[num]] end do;
      for num from 2 by 2 to nops(L) do LL := [op(LL), L[num]] end do;
      return LL
  end if
end proc
> L := [10, 20, 30, 40, 50, 60, 70, 80];
```

$$L := [10, 20, 30, 40, 50, 60, 70, 80] \quad (9)$$

```
> q3(L);
```

$$[10, 30, 50, 70, 20, 40, 60, 80] \quad (10)$$

```
>
> #Q4
> issquare :=proc(L :: list)
  local num, LL, result;
  LL := [ ];
  if irem(nops(L), 2) ≠ 0 then return false;
  else
  result := true;
  end if;
   for num from 1 to nops(L)/2 do
   if L[num] ≠ L[num + nops(L)/2] then return false; end if;
  end do;
```

```
    return true;
    end proc;
issquare := proc(L::list)                                                    (11)
    local num, LL, result;
    LL := [ ];
    if irem(nops(L), 2) <> 0 then return false else result := true end if;
    for num to 1/2*nops(L) do
        if L[num] <> L[num + 1/2*nops(L)] then return false end if
    end do;
    return true
end proc
```

```
> issquare([1, 2, 1, 2]);
                                    true                                     (12)
```

```
> issquare([1, 2, 1, 2, 3]);
                                    false                                    (13)
```

```
> issquare([1, 2, 3, 1, 2, 2]);
                                    false                                    (14)
```

```
>
```

```
> #Q5
```

```
>
```

```
> sumnonprime :=proc(L :: list)
    local num, sum;
    sum := 0;
    for num from 1 to nops(L) do
    if isprime(L[num]) =false then sum := sum + L[num]; end if;
    end do;
    return sum;
    end proc;
sumnonprime := proc(L::list)                                                 (15)
    local num, sum;
    sum := 0;
    for num to nops(L) do
        if isprime(L[num]) =false then sum := sum + L[num] end if
    end do;
    return sum
end proc
```

```
> L1 := [11, 23, 21, 31];
                                L1 := [11, 23, 21, 31]                       (16)
```

```
> sumnonprime(L1);
                                    21                                       (17)
```

```
> L2 := [11, 21, 31, 42];
                                L2 := [11, 21, 31, 42]                       (18)
```

```
> sumnonprime(L2);
                                    63                                       (19)
```