

```

problema diferenciaMaxima (l:[ $\mathbb{R}$ ]) = result :  $\mathbb{R}$ {
  requiere ordenada(l);
  asegura result == l|l|-1 - l0;
}

problema diferenciaMaxima (l:[ $\mathbb{Z}$ ]) = result :  $\mathbb{Z}$  {
  asegura result ∈ l;
  asegura (∄x ← l) long([y|y ← l, y mod x == 0]) > long([h|h ← l, h mod result == 0]) ∧ x ≠ result;
}

problema cantApariciones (l:[T]) = result : [(T, $\mathbb{Z}$ )] {
  requiere |l| > 0;
  asegura (∀x ← result), prm(result) ∈ l ∧ sgd(result) == cuenta(prm(result), l);
  asegura sinRepetidos(result);
}

problema concatMultiple (ls:[[T]]) = result : [T] {
  asegura result == [xi|x ← ls, i ← [0..|x| - 1]];
}

problema mayorSubLista (l:[[ $\mathbb{Z}$ ]]) = result : [ $\mathbb{Z}$ ] {
  requiere |l| > 0;
  asegura mayor(result) == mayor(concatMultiple(l));
  asegura (∃x ← ls)(∀i ← [0..|x| - 1]) resulti == xi ∧ long(result) == long(x);
}

```