

Ejercicio 1

```
const CANT_JUECES = ... ;
type Puntajes = array [1 .. CANT_JUECES] of real;

function puntaje (panel : Puntajes ) : real;

var
    min, max, acum : real;
    i : integer;
begin
    min := panel[1];
    max := panel[1];
    acum := panel[1];
    for i := 2 to CANT_JUECES do begin
        if min > panel[i] then
            min := panel[i];
        if max < panel[i] then
            max := panel[i];
        acum := acum + panel[i];
    end;

    puntaje := (acum - max - min) / (CANT_JUECES - 2);

end;
```

Ejercicio 2

```
function rango (ini, fin : integer ) : ListaInt;
var    i : integer;
       res, nodo : ListaInt;
begin
    res := nil;
    for i := fin downto ini do
        begin
            new(nodo);
            nodo^.elem := i;
            nodo^.sig   := res;
            res := nodo
        end;
    rango := res
end;
```

Ejercicio 3

```
procedure primerRepetido (act: ArrayConTope; var result: Resultado);
var i, j : integer;
begin
    if act.tope = 0 then begin
        result.hayRepetidos := false;
        result.error := noHayDatos;
    end else begin
        i := 1;
        result.hayRepetidos := false;
        result.error := noHayRepetidos;
        while (i < act.tope) and (not result.hayRepetidos) do begin
```

```
    j := i + 1;
  while (j <= act.tope) and (act.elems[i] <> act.elems[j]) do
    j := j + 1;
  if j <= act.tope then begin
    result.hayRepetidos := true;
    result.primerRepetido := act.elems[j]
  end;
  i := i + 1;
end;
end;
end;
```

Ejercicio 4

Entrada: 0 Salida: 0 0 8 17 8

Entrada: 1 Salida: 1 1 9 19 9

Entrada: 2 Salida: 2 2 8 19 8

Entrada: 3 Salida: 3 3 7 19 7

Entrada: 4 Salida: 4 4 4 17 4

Entrada: 5 Salida: 5 5 1 15 1

Entrada: 6 Salida: 6 6 -4 11 -4

Entrada: 7 Salida: 7 7 -9 7 -9

Entrada: 8 Salida: 8 8 -16 1 -16

Entrada: 9 Salida: 9 9 -23 -5 -23
