

## Ejercicio 1

### Parte a)

```
function compararCodigos(cod1, cod2 : Codigo) : Comparacion;

  var i : Integer;

begin
  i := 1;

  while (i <= MaxCod) and (cod1[i] = cod2[i]) do
    i := i+1;

  if i > MaxCod then
    comparoCodigos := igual
  else if cod1[i] < cod2[i] then
    comparoCodigos := menor
  else
    comparoCodigos := mayor

end;
```

### Parte b)

```
procedure menorArtEval(arts : Articulos; var result : Resultado);

  var i, iMenor : Integer;

begin
  i := 1;

  (* busco el primer articulo evaluado *)
  while (i <= arts.tope) and not arts.elems[i].evaluado do
    i := i+1;

  (* el primer articulo evaluado es mi minimo actual *)
  iMenor := i;

  (* a partir de ese articulo recorro todos los demas *)
  (* si habia superado el tope, no entro al for *)
  for i := iMenor to arts.tope do
    if arts.elems[i].evaluado and
      (compararCodigos(artics.arts[i].codigoArt,
        artics.arts[iMenor].codigoArt) = menor) then
      iMenor := i;

  result.encontrado := iMenor <= arts.tope;

  if result.encontrado then
  begin
    result.cod := arts.elems[iMenor].codigoArt;
    result.eval := arts.elems[iMenor].evaluacion
  end

end;
```

### Parte c)

```
procedure enteroCodigo(n : Integer; var cod :Codigo);  
  
var i, cociente : Integer;  
  
begin  
  cociente := n;  
  for i:= MAX_DIG downto 1 do  
    begin  
      cod[i] := cociente mod 10;  
      cociente := cociente div 10;  
    end;  
end;
```

### Parte d)

```
procedure agregarAutor(autor :Codigo; var art :Articulo);  
  
  var p, celda : ListaCodigoAutores;  
  
  begin  
  
    new(celda);  
    celda^.codigoAutor := autor;  
    celda^.siguiente := nil;  
  
    if art.autores = nil then  
      art.autores := celda  
    else  
      begin  
        p := art.autores;  
        while p^.siguiente <> nil do  
          p := p^.siguiente;  
        p^.siguiente := celda  
      end  
    end;  
end;
```