

Examen de Programación 1 - Soluciones

Instituto de Computación - Facultad de Ingeniería

Febrero 2019

Ejercicio 1

```
function menor_o_igual (d1, d2: dato): boolean;  
begin  
  menor_o_igual := (d1.n < d2.n) or ((d1.n = d2.n) and (d1.c <= d2.c))  
end;
```

```
procedure ordenamiento_seleccion (var tabla: TablaDatos);  
var i, idx, mx : integer;  
    swap : dato;  
begin  
  with tabla do  
    begin  
      for i := tope downto 2 do  
        begin  
          mx := 1;  
          for idx := 2 to i do  
            if menor_o_igual (arre[mx], arre[idx]) then mx := idx;  
            swap := arre[mx]; arre[mx] := arre[i]; arre[i] := swap  
          end  
        end  
      end  
    end;  
end;
```

Ejercicio 2

```
procedure promedio (a: Arreglo; var resultado: TalvezReal);  
var cant, i : integer;  
    sum : real;  
begin  
  cant := 0; sum := 0;  
  for i := 1 to MAX do  
    if a[i].definido then  
      begin  
        cant := 1 + cant;  
        sum := sum + a[i].numero  
      end;  
  resultado.definido := cant > 0;  
  if resultado.definido then  
    resultado.numero := sum / cant  
end;
```

```

procedure real_posterior (a: Arreglo; valor: Real; var resultado: TalvezReal);
var idx : integer;
begin
  idx := 1;
  while (idx <= MAX) and (not a[idx].definido or (a[idx].numero <> valor)) do
    idx := 1 + idx;
  repeat idx := 1 + idx
  until (idx > MAX) or a[idx].definido;
  resultado.definido := idx <= MAX;
  if resultado.definido then
    resultado.numero := a[idx].numero
  end;
end;

```

Ejercicio 3

```

procedure partirPrimero (sep: char; var list, resto: Lista);
var p : lista;
begin
  if (list = nil) or (list^.elem = sep) then
    begin
      resto := list; list := nil
    end else
    begin
      p := list;
      while (p^.sig <> nil) and (p^.sig^.elem <> sep) do p := p^.sig;
      resto := p^.sig;
      p^.sig := nil
    end
  end;
end;

```

Ejercicio 4

Nro				
0	-5	0	5	10
1	-3	1	4	9
2	-1	2	3	8
3	1	3	2	7
4	3	4	1	6
5	5	5	0	5
6	7	6	-1	4
7	9	7	-2	3
8	11	8	-3	2
9	13	9	-4	1