Examen de Programación 1
Instituto de Computación - Facultad de Ingeniería
Febrero 2020

Ejercicio 1

Parte a)

```
function largoLista (lista : TipoLista) : integer;
var
   contador : integer;
   p : TipoLista;
begin
   contador:= 0;
   p:= lista;
   while p <> nil do
   begin
      contador:= contador + 1;
      p:= p^.sig;
   end;
   largoLista:= contador;
end;
```

Parte b)

```
function menorLargo (casillas : TipoCasillas) : RangoIndice;
var
   menor, actual : integer;
   ind : RangoIndice;
   j : 1..MAX + 1;
begin
   menor := largoLista (casillas[1]);
   ind := 1;
   j := 2;
   while (menor > 0) and (j <= MAX) do
   begin
      actual := largoLista (casillas[j]);
      if actual < menor then
      begin
         menor := actual;
         ind := j;
      end;
      j := j + 1
   end;
   menorLargo := ind;
end;
```

Parte c)

```
procedure agregaElemento (ele : char; var lista : TipoLista);
var  p, q: TipoLista;
begin
   (* creo nodo, el elemento se agrega siempre *)
   new(p);
```

```
   p^.valor := ele;
   if lista = NIL then
   begin
      p^.sig := NIL ;
      lista := p
   end
   else
   begin
      (* busco el elemento ele en la lista *)
      q := lista;
      while (q^.sig <> NIL) and (q^.valor <> ele) do
         q:= q^.sig;
      (* si esta, lo agrego a continuación si no esta, lo agrego al final *)
      p^.sig := q^.sig;
      q^.sig := p;
   end;
end;
```

Parte d)

```
procedure  enLaDeMenorLargo (ele : char; var casillas : TipoCasillas);
var ind : RangoIndice;
begin
   ind := menorLargo (casillas);
   agregaElemento (ele, casillas[ind]);
end;
```

Ejercicio 2

Parte a)

```
function ordenado (fila:TipoFila):Boolean;
var i : integer;
begin
   i := 1;
   while (i < N) and (fila[i] <= fila[i+1]) do
      i := i + 1;
   ordenado := i = N
end;
```

Parte b)

```
procedure ordenarMatriz (var matriz : TipoMatriz; var cantidad : integer);
var i : 1..N;
begin
   cantidad := 0;
   for i := 1 to N do
      if not ordenado (matriz[i]) then
```

```
      begin
         ordenarFila (matriz[i]);
         cantidad := cantidad + 1
      end
end;
```

Ejercicio 3

```
procedure unir(var a : TipoArregloTope; b : TipoArregloTope);
var aux : 1..N + 1;
begin
   aux := 1;
   if a.tope > 0 then
      while (aux <= b.tope) and (b.arr[aux] < a.arr[a.tope]) do
      begin
         aux := aux + 1;
      end;

   while (a.tope < N) and (aux <= b.tope) do
   begin
      a.arr[a.tope + 1] := b.arr[aux];
      a.tope := a.tope + 1;
      aux := aux + 1;
   end;
end;
```