Programación 1
Segundo Parcial
Instituto de Computación - Facultad de Ingeniería
Julio 2019

Soluciones

Ejercicio 1

Parte a)

```
function cadenasIguales (cad1, cad2 : Cadena) : Boolean;
var i: Integer;
   iguales: Boolean;
begin
   iguales := cad1.largo = cad2.largo;
   i := 1;
   while iguales and (i <= cad1.largo) do
   begin
      iguales := cad1.letras[i] = cad2.letras[i];
      i := i + 1
   end;
   cadenasIguales := iguales
end;
```

```
function cadenasIguales (cad1, cad2 : Cadena) : Boolean;
var i: Integer;
begin
   if cad1.largo = cad2.largo then
   begin
       i := 1;
       while (i <= cad1.largo) and (cad1.letras[i] = cad2.letras[i]) do
           i := i + 1;
       cadenasIguales = i > cad1.largo
   end
   else
       cadenasIguales := false
end;
```

Parte b)

```
procedure buscarPersonaUnica(nombre: Cadena; lista: grupo;
                             var personaEncontrada: persona; var esUnica: Boolean);
var i : Integer;

begin

   (* busco la primera persona con el nombre dado *)
   i := 1;
   while (i <= CANT_PERS) and (not cadenasIguales(arreglo[i].nombre, nombre)) do
      i := i + 1;

   (* encontré una persona, la guardo en el resultado y busco alguna más *)
   if i <= CANT_PERS then
   begin
      personaEncontrada := arreglo[i];
      i := i + 1;
      while (i <= CANT_PERS) and (not cadenasIguales(lista[i].nombre, nombre)) do
         i := i + 1;
     (* si encontré una más, devuelvo FALSE, si no, devuelvo TRUE *)
      esUnica := i > CANT_PERS
   end

    (* no encontré ninguna persona con el nombre dado*)
   else
   begin
      personaEncontrada.generacion := nadie;
      personaEncontrada.nombre.largo := 0;
      esUnica := FALSE
   end
end;
```

Parte c)

```
function  maximos(lista : grupo; var maxEdad: Integer; maxSalario : Real);
(* asumo que el salario es >= 0 *)
var maxE, i : Integer;
    maxS    : Real;
begin
   maxE := -1;
   maxS := -1;
   for i := 1 to CANT_PERS do
      if (arreglo[i].generacion = menor) and (arreglo[i].edad > maxE) then
         maxE := arreglo[i].edad
      else if (arreglo[i].generacion = adulto) and (arreglo[i].salario > maxS) then
         maxS := arreglo[i].salario;
   maxEdad := maxE;
   maxSalario := maxS (* se podría trabajar directo con los parámetros *)
end;
```

Ejercicio 2

```
procedure insertar (var lis1, lis2 : ListaInt; pos : Integer);
var i:Integer;
    p,q : ListaInt;
begin
   p := lis2;
   i := pos;
   while (p^.sig <> nil) and (i > 1) do
   begin
      i := i - 1;
      p := p^.sig
   end;
   if (p^.sig <> nil) then
   begin
      q := ultimo(lis1);
      q^.sig := p^.sig
   end;
   p^.sig := lis1
end;
```

Ejercicio 3

```
function indiceElementoMenor(r: Real): Integer;
var elemento: Real;
    indice: Integer;
begin
   elemento := 0.5;
   indice := 1;      (* como r <= 1, nunca retorna 0, porque A0=1 *)
   while elemento >= r do
   begin
      elemento := elemento/2;
      indice := indice + 1
   end;
   indiceElementoMenor := indice
end;
```