Solución Examen de Programación 1
Instituto de Computación - Facultad de Ingeniería
Julio 2019

Ejercicio 1

a)

```
function ApareceVeces (car : Char; veces : Integer; arreglo : ArregloChars) : Comparacion;
var
   i, cant :  integer;
begin
   i:= 1;
   cant:= 0;

   while (i <= arreglo.tope) and (cant <= veces) do
   begin
      if arreglo.info[i] = car then
         cant:= cant + 1;
      i:= i + 1;
   end;

   if cant < veces then
      ApareceVeces:= menos
   else if cant = veces then
      ApareceVeces:= igual
   else
      ApareceVeces:= mas
end;
```

b)

```
function LargoMaxSecuencia (car : Char; arreglo : ArregloChars) : Integer;
var
   i, cont, maximo : integer;
begin
   i:= 1;
   maximo:= 0;

   while (i <= arreglo.tope) do
   begin
      {avanzar hasta encontrar car}
      while (i <= arreglo.tope) and (arreglo.info[i] <> car) do
         i:= i + 1;

      {comienza secuencia de car, aunque puede ser vacía}
      cont:= 0;
      while (i <= arreglo.tope) and (arreglo.info[i] = car) do
      begin
         cont:= cont + 1;
         i:= i + 1;
      end;

      {actualizar max}
      if cont > maximo then
         maximo:= cont;
   end;
   LargoMaxSecuencia:= maximo
end;
```

Ejercicio 2

a)

```
function notaEst (ci : Integer; grupo : GrupoEst) : Integer;
var i : integer;
begin
   i := 1;
   while (i <= grupo.tope) and (grupo.est[i].cedula <> ci) do
       i:= i+1;
   if (i <= grupo.tope) and (grupo.est[i].estado = finalizado) then
       notaEst := (grupo.est[i].nota)
   else
       notaEst := -1;
end;
```

b)

```
function cantEst (calificacion : rango; grupo : GrupoEst) : integer;
var i, cont :integer;
begin
   cont := 0;
   for i := 1 to grupo.tope do
      if (grupo.est[i].estado = finalizado) and (grupo.est[i].nota >= calificacion) then
          cont := cont +1;
  cantEst := cont;
end;
```

c)

```
procedure subGrupoEst (grupo : GrupoEst; var subgrupo : GrupoEst);
var i : integer;
begin
   subgrupo.tope := 0;
   for i := 1 to grupo.tope do
      if (grupo.est[i].estado = en_curso) then
      begin
         subgrupo.tope := subgrupo.tope +1;
         subgrupo.est[subgrupo.tope] := grupo.est[i];
      end;
end;
```

Ejercicio 3

```
function SumaDespues (k : integer; lista : Lint) : integer;
var
   suma : integer;
   p    : Lint;
begin
   p:= lista;

   {busco primera aparición de k}
   while (p <> nil) and (p^.info <> k) do
      p:= p^.sig;

   if p <> nil then
      {salteo k}
      p:= p^.sig;

   {recorro resto y calculo suma}
   suma:= 0;
   while (p <> nil) do
   begin
      suma:= suma + p^.info;
      p:= p^.sig
   end;

   SumaDespues:= suma;
end;
```