

Examen Diciembre 2019. Soluciones

Ejercicio 1

Con divisiones sucesivas:

```
procedure esPotencia2(num : Natural; var respuesta : Resultado);
var
  k,aux : integer;
begin
  if num = 0 then
    { debo tratar 0 aparte }
    respuesta.exito:= false
  else
    begin
      aux:= num;
      k:= 0;
      { divisiones sucesivas hasta llegar a un impar }
      while aux mod 2 = 0 do
        begin
          k:= k + 1;
          aux:= aux div 2
        end;

        { si el impar es 1 es potencia de 2 }
        respuesta.exito:= aux = 1;
        if respuesta.exito then
          { el exponente es la cantidad de divisiones }
          respuesta.exponente:= k
        end;
      end;
    end;
```

Con multiplicaciones sucesivas:

```
procedure esPotencia2(num : integer; var respuesta : Resultado);
var
  k,aux : integer;
begin
  aux:= 1;
  k:= 0;
  { multiplicaciones sucesivas hasta llegar a num o pasarse }
  while aux < num do
  begin
    aux:= aux * 2;
    k:= k + 1
  end;

  { si queda en num es potencia de 2 }
  respuesta.exito:= aux = num;
  if respuesta.exito then
    { el exponente es la cantidad de multiplicaciones }
    respuesta.exponente:= k
  end;
```

Ejercicio 2

Parte a)

```
procedure trim (var a : tarreglo);
var
  i,j,cont : integer;
begin
  i:= 1;
  { determino la cantida de ceros al principio}
  while (i <= a.tope) and (a.info[i] = '0' ) do
    i:= i + 1;
    cont:= i - 1; {cantidad de ceros}

  if cont > 0 then
  begin
    { desplazo cont posiciones al resto}
    for j:= i to a.tope do
      a.info[j-cont] := a.info[j];

    { ajuste del tope}
    a.tope:= a.tope - cont
  end
end;
```

Parte b)

```
function mayorDistancia (a : tarreglo) : integer;
var maximo, i, cont : integer;
begin
  maximo:= 0;
  i:= 1;

  { avanzo hasta el primer 1 (si hay) }
  while (i <= a.tope) and (a.info[i] = '0') do
    i:= i + 1;

  i:= i + 1; { salteo el posible primer 1 }
  while (i <= a.tope - maximo) do
  begin

    {avanzo hasta el primer 0 (si hay) }
    while (i <= a.tope-maximo) and (a.info[i] = '1') do
      i:= i + 1;

    if i <= a.tope - maximo then
    begin
      { cuento ceros}
      cont:= 0;
      while(i <= a.tope) and (a.info[i] = '0') do
      begin
        cont:= cont + 1;
        i:= i + 1
      end;

      { la primera condición asegura que encontré un 1 }
      if (i < a.tope) and (cont > maximo) then
        maximo:= cont
      end
    end;

    mayorDistancia:= maximo
  end;
```

Ejercicio 3

```
procedure limpiaPolinomio (var p : Polinomio);
var
  aux,l : Polinomio;
begin
  {
    Borro celdas al principio
    Cambia el valor de p
  }
  while (p <> nil) and (p^.mon.coef = 0) do
  begin
    aux:= p;
    p:= p^.sig;
    dispose(aux);
  end;

  l:= p;
  if l <> nil then
  begin
    { Recorrida para borrar celdas que no están al principio.
      Cambia el puntero de la celda anterior
    }
    while l^.sig <> nil do
    begin
      if l^.sig^.mon.coef = 0 then
      begin
        { borrar celda apuntada por l^.sig }
        aux:= l^.sig;
        l^.sig:= aux^.sig;
        dispose(aux)
      end
      else l:= l^.sig
    end
  end
end;
```