Examen de **Fundamentos de Informática**

Grado de Tecnologías Industriales

Escuela de Ingeniería y Arquitectura

Soluciones examen 8 de Septiembre de 2016

Departamento de Informática e Ingeniería de Sistemas

Universidad Zaragoza

Ejercicio 1

[3 puntos]

```
procedure comprimir(var fOriginal: tpFichero; var fComprimido: tpFichero);
var
        valor, nuevovalor, frecuencia: integer;

begin
        reset(fOriginal);
        rewrite(fComprimido);

        if not eof(fOriginal) then begin

                {Lectura de primer valor de referencia}
                read(fOriginal, valor);
                frecuencia := 1;

                while not eof(fOriginal) do begin

                        read(fOriginal, nuevovalor);

                        if (valor = nuevovalor) then
                                frecuencia := frecuencia + 1
                        else begin
                                write(fComprimido, frecuencia, valor);
                                valor := nuevovalor;
                                frecuencia := 1;
                        end;

                end;

                {Al llegar a fin de fichero no escribe el ultimo par: forzar escritura}
                write(fComprimido, frecuencia, valor);

        end;

        close(fOriginal);
        close(fComprimido);
end.
```

```
type
        tpFichero = file of integer;
```

```
procedure descomprimir(var fComprimido: tpFichero; var fOriginal: tpFichero);
var
        valor, frecuencia, i: integer;

begin
        reset(fComprimido);
        rewrite(fOriginal);

        while not eof(fComprimido) do begin
                read(fComprimido, frecuencia, valor);
                for i:=1 to frecuencia do
                        write(fOriginal, valor);
        end;

        close(fComprimido);
        close(fOriginal);
end;
```

Ejercicio 3

[3.5 puntos]

```
const
   SIZE = 5;
   HMAX = 1.561;

type
   tpImagen = array[1..SIZE,1..SIZE] of integer;
   tpAlturas = array[1..SIZE,1..SIZE] of real;

procedure encontrarVehiculos(alturas: tpAlturas; foto: tpImagen);

var
   xBase, yBase: integer;
   i,j: integer;
   dist, max: real;
   cont: integer;

begin

   // Inicialización para el máximo
   xBase := 1;
   yBase := 1;
   max := alturas[yBase,xBase];
   // Búsqueda de la base (el máximo)
   for i:=1 to SIZE do
      for j:=1 to SIZE do
      begin
         if alturas[i,j] > max then
         begin
            xBase := j;
            yBase := i;
            max := alturas[yBase,xBase];
         end;
      end;
   writeln('Encontrada base en ', yBase, '-', xBase, ' con valor de altura ', max:1:3, '.');

   // Busco vehículos y si encuentro uno saco por pantalla su distancia
   cont := 0;
   for i:=1 to SIZE-1 do
      for j:=1 to SIZE-1 do
      begin
         if (foto[i,j] < 100) and (foto[i,j+1] < 100) and (foto[i+1,j] < 100) then
         begin
            dist := sqrt((xBase - j)*(xBase - j)+(yBase - i)*(yBase - i));
            writeln('Encontrado vehiculo en ', i, '-', j, '. Distancia a la base: ', dist:1:2);
            cont := cont + 1;
         end;
      end;
   writeln('Se han encontrado un total de ', cont, ' vehiculos.');
   readln;
   readln

end;
```

Ejercicio 2 [3.5 puntos]

```
program censura;

type
        tpPalabra = record
                letras : array[1..30] of char;
                n : Integer;
        end;

        tpCensuradas = record
                palabras : array[1..500] of tpPalabra;
                n : Integer;
        end;


procedure cargarPalabraCensurada(var t: text; var palabra: tpPalabra);
begin
        palabra.n := 0;
        while not eoln(t) do begin
                palabra.n:=palabra.n+1;
                read(t,palabra.letras[palabra.n]);
        end;
        readln(t);
end;

procedure cargarCensuradas(var censuradas : tpCensuradas);
var t:text;
begin
        assign(t,'censura.txt');
        reset(t);
        censuradas.n:=0;
        while not eof(t) do begin
                censuradas.n:=censuradas.n+1;
                cargarPalabraCensurada(t,censuradas.palabras[censuradas.n]);
        end;
        close(t);
end;

function aMinuscula(c : char) : char;
{ Pre: c es un caracter alfabetico (no es cifra ni otro tipo de caracter) }
begin
        if (c>='A') and (c<='Z') then
                aMinuscula := chr(ord(c)-ord('A')+ord('a'))
        else
                aMinuscula := c;
end;

function palabrasIguales(p1,p2 : tpPalabra) : boolean;
var i : Integer; iguales : boolean;
begin
        if (p1.n <> p2.n) then palabrasIguales:=false
        else begin
                i:=1; iguales := true;
                while (i<=p1.n) and iguales do begin
                        iguales:= aMinuscula(p1.letras[i]) = aMinuscula(p2.letras[i]);
                        i:=i+1;
                end;
                palabrasIguales := iguales;
        end;
end;



function esCensurada(palabra : tpPalabra; censuradas : tpCensuradas) : boolean;
var i : Integer; censurada : boolean;
begin
        i:=1; censurada := false;
        while (i<=censuradas.n) and (not censurada) do begin
                censurada := palabrasIguales(palabra, censuradas.palabras[i]);
                i := i+1;
        end;
        esCensurada := censurada;
end;

procedure escribirPalabra(var t: text; palabra : tpPalabra);
var i : Integer;
begin
        for i := 1 to palabra.n do write(t, palabra.letras[i]);
end;

var
        palabra : tpPalabra;
        censuradas : tpCensuradas;
        text_in  : text;
        text_out : text;
        c : char;
begin
        cargarCensuradas(censuradas);
        assign(text_in,'entrada.txt');
        assign(text_out,'salida.txt');
        reset(text_in);
        rewrite(text_out);

        while not eof(text_in) do begin
                palabra.n := 0;
                while not eoln(text_in) do begin
                        read(text_in, c);
                        if ((c>='A') and (c<='Z')) or ((c>='a') and (c<='z')) then begin
                                palabra.n := palabra.n+1;
                                palabra.letras[palabra.n] := c;
                        end else begin
                                if (palabra.n > 0) then begin { estamos acabando una palabra }
                                        if esCensurada(palabra, censuradas) then write(text_out, 'XXX')
                                        else escribirPalabra(text_out, palabra);
                                        palabra.n:=0;
                                end;
                                write(text_out,c); { escribimos todos los caracteres no correspondientes a u
                        end;
                end;
                { En los saltos de linea hacemos lo mismo que cuando nos encontramos un caracter no alfabeti
                if (palabra.n > 0) then begin { estamos acabando una palabra }
                        if esCensurada(palabra, censuradas) then write(text_out, 'XXX')
                        else escribirPalabra(text_out, palabra);
                        palabra.n:=0;
                end;
                readln(text_in);
                writeln(text_out); { escribimos el salto de linea }
        end;
        close(text_in);
        close(text_out);
end.
```

```
program censura;

type
        tpPalabra = record
                letras : array[1..30] of char;
                n : Integer;
        end;


procedure cargarPalabraCensurada(var t: text; var palabra: tpPalabra);
begin
        palabra.n := 0;
        while not eoln(t) do begin
                palabra.n:=palabra.n+1;
                read(t,palabra.letras[palabra.n]);
        end;
        readln(t);
end;

function aMinuscula(c : char) : char;
{ Pre: c es un caracter alfabetico (no es cifra ni otro tipo de caracter) }
begin
        if (c>='A') and (c<='Z') then
                aMinuscula := chr(ord(c)-ord('A')+ord('a'))
        else
                aMinuscula := c;
end;

function palabrasIguales(p1,p2 : tpPalabra) : boolean;
var i : Integer; iguales : boolean;
begin
        if (p1.n <> p2.n) then palabrasIguales:=false
        else begin
                i:=1; iguales := true;
                while (i<=p1.n) and iguales do begin
                        iguales:= aMinuscula(p1.letras[i]) = aMinuscula(p2.letras[i]);
                        i:=i+1;
                end;
                palabrasIguales := iguales;
        end;
end;

function esCensurada(palabra : tpPalabra; var censuradas : text) : boolean;
var censurada : boolean; p2 : tpPalabra;
begin
        reset(censuradas);
        censurada := false;
        while (not eof(censuradas)) and (not censurada) do begin
                cargarPalabraCensurada(censuradas,p2);
                censurada := palabrasIguales(palabra, p2);
        end;
        esCensurada := censurada;
end;



function esCensurada(palabra : tpPalabra; var censuradas : text) : boolean;
var censurada : boolean; p2 : tpPalabra;
begin
        reset(censuradas);
        censurada := false;
        while (not eof(censuradas)) and (not censurada) do begin
                cargarPalabraCensurada(censuradas,p2);
                censurada := palabrasIguales(palabra, p2);
        end;
        esCensurada := censurada;
end;

procedure escribirPalabra(var t: text; palabra : tpPalabra);
var i : Integer;
begin
        for i := 1 to palabra.n do write(t, palabra.letras[i]);
end;

var
        palabra : tpPalabra;
        censuradas : text;
        text_in  : text;
        text_out : text;
        c : char;
begin
        assign(text_in,'entrada.txt');
        assign(text_out,'salida.txt');
        assign(censuradas,'censura.txt');
        reset(text_in);
        rewrite(text_out);

        while not eof(text_in) do begin
                palabra.n := 0;
                while not eoln(text_in) do begin
                        read(text_in, c);
                        if ((c>='A') and (c<='Z')) or ((c>='a') and (c<='z')) then begin
                                palabra.n := palabra.n+1;
                                palabra.letras[palabra.n] := c;
                        end else begin
                                if (palabra.n > 0) then begin { estamos acabando una palabra }
                                        if esCensurada(palabra, censuradas) then write(text_out, 'XXX')
                                        else escribirPalabra(text_out, palabra);
                                        palabra.n:=0;
                                end;
                                write(text_out,c); { escribimos todos los caracteres no correspondientes a una palabra }
                        end;
                end;
                { En los saltos de linea hacemos lo mismo que cuando nos encontramos un caracter no alfabetico }
                if (palabra.n > 0) then begin { estamos acabando una palabra }
                        if esCensurada(palabra, censuradas) then write(text_out, 'XXX')
                        else escribirPalabra(text_out, palabra);
                        palabra.n:=0;
                end;
                readln(text_in);
                writeln(text_out); { escribimos el salto de linea }
        end;
        close(text_in);
        close(text_out);
        close(censuradas);
end.
```