

JUEGO DE LA ORACION

Solución acordada con observaciones

Observaciones al código hechas por los alumnos:

- Mantener un orden con espaciados entre las frases y las respuestas para que uno cuando lee no se pierda o se confunda, más que nada organización. También tendría que encontrar la forma de manejar una estructura que controle el juego de manera que se pueda llevar a cabo más de un solo juego sin tener que reiniciar el juego o los jugadores.
- Al finalizar el juego no se alcanza a ver quien es el ganador y sale sin ningún aviso un consejo imprimir una condición de finalización del juego como Writeln('presione enter para finalizar');
- Luego el código Readln; hace que el operador tenga que presionar enter o cualquier tecla de modo que se finaliza la ejecución del programa.
- Me gusta la forma como enuncia la pregunta, las afirmaciones y todo lo referido a la impresión del programa, es llamativo y le da un buen aspecto al programa.

```
program entrega1;
```

```
const
```

```
    esp = ' '; {QUE FUNCION CUMPLE??}
```

```
type
```

```
    longText = string;
```

```
    texto = string[20];
```

```
    jug = ^nodo1;           {lista circular de jugadores}
```

```
    nodo1 = record
        nombre: texto;
        sig: jug;
    end;
```

```
    fra = ^nodo2;           {lista simple de palabras}
```

```
    nodo2 = record
        palabra: texto;
        sig: fra;
    end;
```

```
procedure cargarJugadores(VAR pri: jug; resp: string; VAR cant: integer);
```

```
var
```

```
    nuevo: jug;
```

```
begin
```

```
    while (resp = 's') do begin
```

```

        cant:= cant +1;
        new(nuevo);
        writeln('>> Nombre del jugador:');
        readln(nuevo^.nombre);
        if (pri = nil) then begin           {agrego el primer nodo a la lista}
            pri:= nuevo;
            pri^.sig:= nuevo
        end
        else begin
            nuevo^.sig:= pri^.sig;
            pri^.sig = nuevo;

        end;
        writeln('>>> Desea agregar otro jugador? [s/n]:');
        readln(resp);
    end;
    if (resp = 'n') then                   {si responde que no, sale}
        writeln('## Carga de jugadores finalizada.');
```

{OBSERVAR QUE ES UNA CREACION DE LISTA CIRCULAR PURA}

```

procedure eliminarjugador (var pri:jug; var ant:jug; VAR cant: integer);
    var
        act:jug;
    begin
        writeln ('>> Jugador/a ',pri^.nombre, ' eliminado. ');
        cant:= cant-1;
        act:=pri;
        pri:=pri^.sig;
        ant^.sig:=pri;
        dispose (act);
    end;
```

```

procedure cargarPalabra(palabra: texto; VAR pri: fra);
    var
        nuevo: fra;
    begin
        new(nuevo);
        nuevo^.palabra:= palabra;
        if (pri = nil) then                 {agrego el nuevo nodo a la lista}
            pri:= nuevo
        else begin
```

```
                pri^.sig:= nuevo;
                nuevo^.sig:= nil;
                end;
end;
```

```
procedure comparar(cantPal: integer; frase:fra; VAR error: boolean);
var
    pal: texto;
    increm: integer;
begin
    readln(pal);
    increm:=1;
    while (pal = frase^.palabra) and (increm < cantPal) do begin
        increm:= increm +1;
        if (cantpal > 1) then begin
            frase:= frase^.sig;
            readln(pal);
        end;
    end;
    if (frase^.palabra <> pal) then begin
        writeln('>> FRASE ERRONEA!');
        error:= true;
    end
    else
        if (cantPal = increm) then
            error:= false;
end;
```

```
procedure comenzarjuego (VAR pri:jug;respuesta: string; VAR cant: integer; VAR frase: fra);
var
    palabraNue: texto;
    jugAnt:jug;
    cantPal: integer;
    error: boolean;
begin
    cantPal:= 0;
    error:= false;
    if (cant > 1) then begin
        writeln(pri^.nombre,' es tu turno!');
        writeln('>>>> Ingrese una palabra para comenzar a jugar: ');
    end;
```

```

        readln(palabraNue);
        cantPal:= cantPal +1;
        cargarPalabra(palabraNue, frase);
        jugAnt:=pri;
        pri:= pri^.sig;
        writeln('>>>> Deseas continuar jugando, ',pri^.nombre,'? [s/n]:');
        readln(respuesta);
        end;
    while (cant > 1) do begin
        if (respuesta = 's') then begin
            writeln('>>>> Ingrese la palabra/frase anterior:');
            writeln('Tip: En caso de ser una frase, por favor, ingrese palabra
por palabra, presionando [Enter] para cada una. ');
            comparar(cantPal,frase,error);
            if (error = false) then begin {SE PODRÍA REALIZAR UNA
FUNCTION QUE DEVUELVA EL VALOR TRUE O FALSE EN EL CASO QUE
CORRESPONDA}
                writeln('Ingrese la nueva palabra que desea anadir a la
frase:');

                readln(palabraNue);
                cargarPalabra(palabraNue,frase);
                jugAnt:=pri;
                pri:=pri^.sig;
                end;
            end;
            write('>>>> Deseas continuar jugando, ',pri^.nombre,'? [s/n]: ');
            readln(respuesta);
            if (respuesta = 'n') then
                eliminarJugador(pri,jugAnt,cant);
            while (respuesta = 's') and (cant > 1) do begin
                writeln('>>>> Ingrese la frase anterior para continuar: ');
                comparar(cantPal,frase,error);
                if (error = false) then begin
                    writeln('Ingrese la nueva palabra que desea anadir a la
frase:');

                    readln(palabraNue);
                    cantpal:=cantpal +1;
                    cargarPalabra(palabraNue,frase);
                    jugAnt:= pri;
                    pri:= pri^.sig;
                    writeln(pri^.nombre,', es tu turno!');
                    end
                else begin

```

```

        writeln('---> ',pri^.nombre,' se confundio! Ha sido
eliminado/a');
        eliminarJugador(pri,jugAnt,cant);           {ya me lo
devuelve en la sig posicion}
        end;
        if (cant > 1) then begin
            write('>>>> ¿Deseas continuar jugando, ',pri^.nombre,'?
[s/n]: ');
            readln(respuesta);
            while (respuesta = 'n') and (cant > 1) do begin
                eliminarJugador(pri,jugAnt,cant);
                write('>>>> ¿Deseas continuar jugando,
',pri^.nombre,'? [s/n]: ');
                readln(respuesta);
            end;
        end;
    end;
end;

```

```

var
    players:jug;
    frase: fra;
    respuesta: char;
    cantidad: integer;

begin
    players:=nil;
    frase:= nil;
    writeln('- - JUEGO DE LA ORACION v2.0.0 - -');{PODRIA UTILIZAR UNA
ESTRUCTURA DE CONTROL POR SI UNA VEZ FINALIZADA UNA PARTIDA VOLVER A
JUGAR SIN NECESIDAD DE ABRIR NUEVAMENTE EL .EXE}
    writeln('>>>> Desea cargar un jugador? [s/n]'); {MODULARIZAR}
    readln(respuesta);
    cantidad:=0;
    if (respuesta = 's') then           {SERIA MAS EFICIENTE MODULARIZAR}
        cargarjugadores (players,respuesta,cantidad);
    if (respuesta = 'n') then
        writeln('NO SE CARGARON JUGADORES, IMPOSIBLE JUGAR!');
    if (cantidad = 1) then
        writeln('NO SE PUEDE JUGAR CON UNA SOLA PERSONA!');
    if (players <> nil) and (cantidad > 1) then begin

```

```
writeln('---> ¿Desea comenzar a jugar? [s/n]: ');
readln(respuesta);
if (respuesta = 's') then begin
    comenzarjuego(players,respuesta,cantidad,frase);
    if (cantidad = 1) then
        writeln ('-----EL GANADOR ES ', players^.nombre,
'.FELICITACIONES -----');
        end;
    if (respuesta = 'n') then begin
        writeln('JUEGO FINALIZADO. '); {UNA VEZ QUE FINALIZA EL JUEGO,
SALE SIN SIQUIERA PODER ALCANZAR A VER QUIEN GANO, DEBERIA COLOCAR UNA
ESTRUCTURA QUE HAGA UN PAUSA ANTES DE CERRAR EL PROGRAMA}
        end;
    end;
end.
```