UNIVERSIDADE DA CORUÑA

FACULTADE DE INFORMÁTICA
Departamento de Computación
Estructura de Datos e da Información

Práctica 0a: Compilador y Punteros

Entorno de trabajo:

- El sistema operativo: Linux Ubuntu o WindowsXP Pro.
- Los editores: Ubuntu (Nano, emacs, Xfte, Vim, gedit), WindowsXP Pro (Notepad, WordPad, Notepad2[1])
- El compilador: FreePascal[2] (local).
- Modo de compilación en línea:
  - Linux: apertura de terminal (`Menu->Accesorios`)
  - WinXP: apertura de ventana de comandos (`Inicio->Accesorios->Simbolo de Sistema` o bien `Inicio->Ejecutar->cmd.exe`
  - Teclear: `fpc <fuente>.pas`.
- ¿Entornos de desarrollo: IDE FreePascal (WinXP), etc...?

Ejercicios:

1. Distinguir entre errores de compilación y errores de ejecución con punteros. Para ello asumir las siguientes declaraciones:

```
var
    x: integer;
    P1, P2: ^integer;
    Q1, Q2: ^real;
```

   Construir un programa principal añadiendo de cada vez sólo una de las siguientes sentencias para probar si son correctas.

   a) `writeln(P1);`

   b) `P1:= Q1;`

   c) `if P1^ = nil then Q1:= Q2;`

   d) `readln(P1^)`

   e)
```
readln(P1^);
writeln('El valor es ', P1^);
```

   f) `new(X)`

   g) `Q1^:=P1^`

[1] Disponible en `http://www.flos-freeware.ch/zip/notepad2.zip`
[2] Disponible en `http://www.freepascal.org`

h)

```
begin
    P1^:= 17;
    new(P1);
end
```

2. Con la misma declaración de variables del ejercicio anterior, compilar y ejecutar el siguiente código:

```
new(p1);
p1^:= 123;
writeln ('p1 es ', p1^);
dispose(p1);
p1:= nil;
writeln ('p1 es ', p1^);
```

   ¿Por qué se produce un error de ejecución?

3. El siguiente programa maneja un tipo de dato `tEstudiante` como un registro con dos campos: nombre y teléfono, y un tipo de dato `pNodo` puntero a registro.El objetivo es:

   - Pedir por teclado los datos de dos estudiantes y almacenarlos a través del puntero.
   - Comparar los dos registros (campo a campo) y determinar si son o no iguales.

   Rellenar las operaciones `CrearNodo`, `MeterDatos` y `Comparar` de acuerdo a la descripción que las acompaña.

```
program prueba;
type
    pNodo = ^tEstudiante;
    tEstudiante = record
        nombre: string;
        tlfno: integer;
    end;
var
    P, Q: ^tEstudiante;

Procedure CrearNodo (var P: PNodo);
{Crea la variable dinámica asociada a P}
begin
   ...
end;

Procedure MeterDatos (var D: tEstudiante);
{Pide por teclado nombre y teléfono y lo almacena en D}
begin
   ...
end;

Function Comparar (P1, P2: pNodo): boolean;
{Compara el contenido de las variables dinámicas asociadas}
{a los punteros P1 y P2 y devuelve Verdadero o Falso}
begin
  ...
end;

begin
    CrearNodo(P); CrearNodo(Q);
    MeterDatos(P^); MeterDatos(Q^);
    if Comparar (P,Q)
```

```
    then writeln ('son iguales')
    else writeln ('son diferentes');
    dispose(P); dispose(Q);
end.
```

Soluciones

1. *a*) `writeln(P1);` Error de compilación: las variables de tipo puntero no se "leen" o "escriben" en sentencias de este tipo.
   `Error: Can't read or write variables of this type`

   b) `P1:= Q1;` Error de compilación: las variables son de tipos distintos.
   `Error: Incompatible types: got ^S64REAL expected ^SMALLINT`

   c) `if P1^ = nil then Q1:= Q2;` `P1` es una variable de tipo puntero; no así `P1^`, la variable dinámica asociada, de tipo entero. La comparación no se puede realizar.

   d) `readln(P1^)` Error de ejecución. `Runtime error 216 at $0040105E`

   e)
   ```
   readln(P1^);
   writeln('El valor es ', P1^);
   ```
   No es correcto. Deberíamos crear previamente la variable dinámica (`New(P1)`).

   f) `new(X)` X no es una variable de tipo puntero.
   `Error: pointer type expected, but got SMALLINT`

   g) `Q1^:=P1^` Correcto sintácticamente pero las variables apuntadas por Q1 y P1 no existen (sería preciso utilizar previamente llamadas a `New(Q1)` y `New(P1)`).

   h)
   ```
   begin
      P1^:= 17;
      new(P1);
   end
   ```
   No hay error de compilación, sino de ejecución puesto que no se ha hecho una reserva previa para el puntero P1, y su valor es indefinido. El orden de las sentencias está intercambiado.

2. Se produce un error de ejecución al tratar de acceder a `P1^` después de hacer el `dispose(P1)`.

3.
```
program prueba;
type
    pNodo = ^tEstudiante;
    tEstudiante = record
        nombre: string;
        tlfno: integer;
    end;
var
    P, Q: ^tEstudiante;

Procedure CrearNodo (var P: PNodo);
begin
   New(P);
end;

Procedure MeterDatos (var D: tEstudiante);
begin
  with d do begin
        write('Intro nombre '); readln (nombre);
        write('Intro tlfno '); readln (tlfno);
  end;
```

```
end;

Function Comparar (P1, P2: pNodo): boolean;
begin
  Comparar:= (p1^.nombre = p2^.nombre) and (p1^.tlfno = p2^.tlfno);
end;

begin
    CrearNodo(P); CrearNodo(Q);
    MeterDatos(P^); MeterDatos(Q^);
    if Comparar (P,Q)
    then writeln ('son iguales')
    else writeln ('son diferentes');
    dispose(P); dispose(Q);
end.
```