Departamento de Informática
Universidad de Valladolid

Ingeniería Técnica de Informática de Sistemas y Gestión
Examen de Programación I
Convocatoria Ordinaria, 13 de Febrero del 2004

Apellidos_____

Nombre_____ **Grupo** _____

- Duración del examen: 3 horas
- Poner el nombre y apellidos en todas las hojas del examen.

- Realizar cada problema en páginas diferentes
- Se valorará la presentación
- Se valorará la adecuación de las estructuras utilizadas al problema a resolver.

1. *(2 puntos)* Dado el siguiente programa:
   a. Especifique qué errores contiene (de compilación, ejecución...) y por qué son errores, qué cosas no se deben hacer (aunque no den error) y por qué.
   b. Especifique, para el programa anterior, el ámbito de los subprogramas que contiene, detallando quiénes los pueden utilizar.

```
1     program examen (input, output, f, g);
2           const x = 7; y = 19;
3           type  tmat = array (1..x) of real;
4           procedure p1(var x: integer; y: real; var m: tmat);
5                 var x: integer; z:real;
6                 begin
7                       x:= z*x;
8                       m:= (x+y)div z;
9                       if f1 then
10                            x:=x+1;
11                end;{p1}
12          function f1 ():boolean;
13                 begin
14                      if x>y then
15                            f1:=true
16                      else
17                            f1:=false;
18                end; {f1}
19          var a,b:integer; f,g:file of tmat; r:tmat;
20          begin
21                readln(a,b,r);
22                p1(a,b,r);
23                writeln (a,b);
24                while not eof(f) do
25                      begin
26                            readln(f, r);
27                            if f1 then
28                                  write(g, r);
29                      end
30          end.{examen}
```

Departamento de Informática
Universidad de Valladolid

Ingeniería Técnica de Informática de Sistemas y Gestión
Examen de Programación I
Convocatoria Ordinaria, 13 de Febrero del 2004

2. *(2 puntos)* Un número entero positivo puede "reducirse a un dígito" a base de sumar sus dígitos y realizar el mismo proceso con el resultado una serie de veces. Por ejemplo, 32767 daría 25 (=3+2+7+6+7) y luego 7 (=2+5), quedando reducido a 7.
   Escríbase una función en Pascal que devuelva la reducción a un dígito de un entero positivo en el rango `integer`, documentando las condiciones bajo las cuales sería correcto su uso (precondición).
   a) Usando recursión.
   b) Sin usar recursión.

   En ambos casos puede resolverse el problema mediante combinación de varias funciones si se considera adecuado, con los mismos condicionantes (recursión o no).

3. *(2 puntos)* Sea una lista enlazada apuntada por un puntero P y un nodo de dicha lista apuntado por otro puntero R. Desarrollar un subprograma Pascal que intercambie, dentro de la lista, la posición del nodo apuntado por R con la del nodo siguiente a él si lo hubiera. Si no lo hubiera, la lista no debe ser modificada. Supóngase (como precondición) que la lista no está vacía y R apunta efectivamente a un elemento existente en la lista. Especifíquese la definición de tipo empleada.
   (Ejemplo: una lista con valores 10,20,30,40,50, 60, donde R apunte al nodo de valor 40, debe quedar en la forma 10, 20, 30, 50, 40, 60)

4. *(4 puntos)* Realizar el diseño modular y la codificación en Pascal de un programa para resolver una sopa de letras de tamaño 20 x 20 con la especificación que sigue:
   Entrada:
   - Un fichero ("sopa.dat") con un único elemento de tipo matriz de 20x20 caracteres, con la sopa.
   - Un fichero de texto ("palabras.txt"), con las palabras a buscar, a razón de una palabra por línea. Cada palabra tiene a lo sumo 10 caracteres y no es vacía.

   Salida:
   - Un fichero de registros ("salida.dat"), con tantos registros como palabras haya en "palabras.txt", en cada uno de los cuales debe estar toda la información asociada con la palabra: fila y columna inicial donde se encuentra en la sopa, dirección y sentido.

   Observaciones:
   - Considérense solamente las direcciones y sentidos siguientes:
     Horizontal de izquierda a derecha (→)
     Horizontal de derecha a izquierda (←)
     Vertical de arriba a abajo (↓)
     Diagonal de arriba-izquierda hacia abajo-derecha (↘)
   - Condiciones de entrada (precondiciones): el fichero "sopa.dat" no está vacío y es correcto. Todas las palabras de "palabras.txt" aparecen en la sopa.
   - La función `POS(subcad,cad:STRING):INTEGER`; devuelve la posición en la que empieza `subcad` dentro de `cad`. Si no está, devuelve 0.