

NOMBRE.....

NÚM. de MATRÍCULA..... GRUPO.....

Examen de Informática. Febrero 2008

Instrucciones

- El examen consta de **diez** preguntas, cada una de las cuales puntuará **cero** o **un** punto.
- Se calificará con **un punto** si la respuesta correcta se indica en la forma **más simple**.
- La duración total del examen será de **dos horas y media**

1. Dada la representación **hexadecimal** de dos valores numéricos enteros, completar las siguientes codificaciones en los formatos de representación **decimal** y **binario** indicados en cada caso.

Codificación hexadecimal:	+A6	Codificación decimal:	
Codificación binaria con el mínimo nº de bits y Complemento a 2			
Codificación hexadecimal:	-A6	Codificación decimal:	
Codificación binaria con el mínimo nº de bits y Complemento a 2			

2. Completar el programa `calmedia` para que durante su ejecución, **asigne** valor a una serie indeterminada de calificaciones numéricas reales introducidas por el usuario por **teclado** y, posteriormente, **calcule** y **visualice** por pantalla la **calificación media** de todas ellas. *Notas:* El usuario indica el final de la serie al introducir un valor **negativo**. Dicho valor **no** se debe contabilizar para el cálculo de la media. Tampoco se deben contabilizar las calificaciones **superiores** a 10 para el cálculo de la media. El programa **no** debe generar un **error** de ejecución si el número de calificaciones es **nulo** (en este caso, la media es 0).

```
program calmedia;
```

```
writeln('La calificación media es: ',cm:10:2);  
end.
```

3. Completar el procedimiento `rangol` para que genere y **devuelva** como resultado una matriz con **rango** igual a **1** de dimensiones $m \times n$. Los valores de los elementos de la matriz deberán ser **todos distintos**. En el cuadro de la derecha se debe indicar los **valores** de los elementos de la matriz generada.

```
const m=3; n=4;
type matriz=array[1..m,1..n] of integer;
procedure rangol
```


```
end;
var a:matriz; i,j:byte;
begin
rangol(m,n,a); writeln('La siguiente matriz tiene rango 1: ');
for i:=1 to m do begin for j:=1 to n do write(a[i,j]:5); writeln end
end.
```

4. Considerando que $\int_a^b f(x) \cdot dx = \int_a^m f(x) \cdot dx + \int_m^b f(x) \cdot dx$, siendo $m = \frac{a+b}{2}$, completar la función **recursiva** `intdef` para que calcule de forma aproximada la **integral definida** de una función $f(x)$ mediante la división en **dos subintervalos**, definidos por el valor de m , del intervalo $[a, b]$ hasta que sea lo bastante pequeño ($|b-a| < \varepsilon$). En este caso se estima que $\int_a^b f(x) \cdot dx \cong (b-a) \cdot f(m)$

```
function f(x:real):real;
begin { Definicion de la funcion f(x) que no es necesario implementar } end;
function intdef(a,b:real; e:real):real;
```

```
end;
```

5. Dadas dos matrices de números enteros de $M \times N$ elementos almacenados de forma **correlativa** por filas en sendos archivos de disco (`mat1.dat` que almacena datos de tipo `integer` y `mat2.txt` en formato ASCII), completar el siguiente programa para que devuelva el **número** de elementos **iguales (mismos índices y valor)** en ambas matrices. A la derecha se muestra un ejemplo de contenidos de los archivos para los que el programa muestra por pantalla el valor de 2 (elementos iguales).

mat1.dat

4	1	-1	5	1	6	0	9	7
---	---	----	---	---	---	---	---	---

 eof

mat2.txt

8	-1	3	#13	#10	5	2	-3	#13	#10	0	0	3
---	----	---	-----	-----	---	---	----	-----	-----	---	---	---

 eof

```
type enteros = file of integer;
```

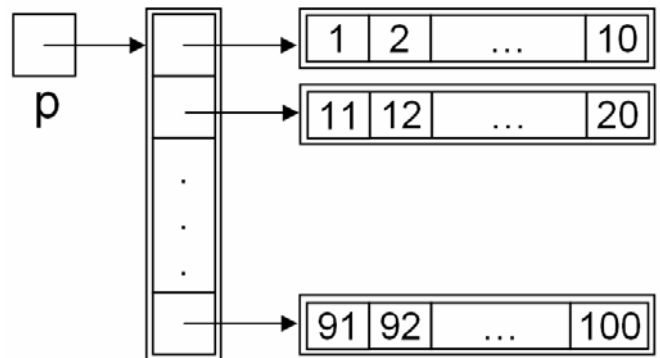
```
writeln('El numero de elementos iguales es: ',k);
end.
```

6. Completar la función *moda* para que devuelva la **moda** de los valores de un **archivo** en disco que almacena valores enteros entre 1 y 100 y cuyo **nombre** se da como **parámetro** de la rutina. *Notas:* En estadística se define *moda* al **valor** que **más veces** aparece **repetido** en un conjunto de valores. Si el archivo está vacío la función debe devolver el valor 0. Si existen dos o más valores con la máxima frecuencia entonces la función debe devolver el menor de los valores.

```
type valor=1..100; enteros = file of valor;
  vector = array[valor] of word;
function moda
```

end;

7. Completar las **declaraciones** necesarias para definir una **estructura** de datos que contiene un *apuntador* que apunta a un vector de *n* elementos de tipo *puntero* que apuntan cada uno de ellos a un vector de *n* enteros. Se debe también completar el procedimiento que genere esa **nueva** estructura dinámica con los valores de la figura.



```
const n=10;
type
```

```
procedure darvalores
```

```
var p:ptr;
begin
darvalores(p);
. . .
```

8. Construir una **unidad** cadenas que declare en la zona pública e implemente un procedimiento modificar que **reemplace** un **carácter** por otro dentro de una **cadena**. Por ejemplo, si *s* es una variable de tipo `string` que toma el valor `'Valentin tiene tilde'`, entonces tras la llamada a `modificar(s, 'i', 'o');`, se cambia el valor de *s* a `'Valenton toene tolde'`.

9. Construir una función que indique si **todos** los valores de una **lista** dinámica simplemente enlazada de números enteros son **distintos**. En este caso la función devuelve el valor `true` y en caso contrario devuelve `false`. El parámetro formal *p* de tipo `ptr` de la función indica la dirección de memoria del primer elemento de la lista. Si la lista está **vacía** la función debe devolver el valor `true`.

```
type ptr = ^elemento;
  elemento = record
    dato: integer;
    sig: ptr;
  end;
```

10. Construir una rutina que **elimine** y **libere** el espacio en memoria correspondiente al **último** elemento de una lista dinámica simple **circular**. La dirección de memoria del primer elemento de la lista se debe dar como parámetro de la rutina. *Notas:* Debe emplearse la declaración de tipos de datos del ejercicio anterior. Debe considerarse el caso de lista **vacía** (no hace nada) y de lista con un **único** elemento (deja la lista original vacía).