

NOMBRE.....

NÚM. de MATRÍCULA..... GRUPO.....

Informática. Examen Final. Septiembre 2005

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**.

1. Representar exactamente el número **-5,50** en formato de coma o punto fijo y en formato de coma o punto flotante **siempre en complemento a 2** y utilizando 8 bits en ambas representaciones. Se pueden repartir en cada caso los 8 bits en la forma mas adecuada para representar la parte entera y decimal en el caso de punto fijo y la mantisa y el exponente en el caso de punto flotante en formato normalizado.

Punto fijo:

Punto Flotante:

2. Dado el siguiente programa:

```
type rec=record
  i:byte;
  j:array[1..2,1..3] of integer;
  s:string[11]
end;
var r1:rec; i:integer;
begin
  r1.i:=1;
  r1.s:='123456';
  for i:=1 to 2 do r1.j[i,i]:=i+1;
  write(r1.i,r1.s[2],r1.j[i,i])
end.
```

Indicar la salida por pantalla al ejecutarse:

3. Dados los vectores v_1 y v_2 de n coordenadas completar el procedimiento `intercalar` para que éste construya un **nuevo vector** v_3 que esté compuesto por los componentes de los vectores v_1 y v_2 intercalados. El vector v_3 deberá tomar un elemento de v_1 y otro de v_2 y así hasta incluir todos sus elementos.

```
const n=10;
type vector1=array [1..n] of integer;
vector2=array[1..2*n] of integer;
var v1,v2:vector1;v3:vector2;
i:integer;
procedure intercalar(v1,v2:vector1; var v3:vector2);
  var i:integer;
  begin
    for i:=1 to 2*n do begin
```

Por ejemplo:

Si v_1 es (1,2,3,4,5,6,7,8,9,10)

y v_2 es (2,4,6,8,10,12,14,16,18,20)

El vector v_3 deberá ser:

(1,2,2,4,3,6,4,8,5,10,6,12,7,14,8,16,9,18,10,20)

end;

end;

4. Escribir un programa que **genere** dos vectores v1 y v2 de números enteros aleatorios **ordenados** de menor a mayor.

```
const n=5;
type vec=array [1..n]of integer;
var v1,v2:vec;
i:integer;
begin
randomize;
v1[1]:=1;v2[1]:=1;
for i:=2 to n do begin
```

Los vectores v1 y v2 generados podrían ser:
v1: (1,2,5,7,12)
v2: (1,5,8,10,15)

end

end.

5. Se define el **valor promedio** de un número entero, al número que se obtiene de sumar sus dígitos de posición impar y restar sus dígitos de posición par (la unidad se considera en posición 1). Completar la función **recursiva** promedio para que devuelva el valor promedio de un entero positivo dado como parámetro. Por ejemplo, el valor promedio de 385147 es 4 ya que $7-4+1-5+8-3=4$.

```
function promedio(n:longint):shortint;
begin
```

end;

6. Dado el siguiente fichero de números enteros 'ent1' de 8 registros:

3	5	6	2	8	9	1	7
---	---	---	---	---	---	---	---

 eof

Completar el programa para que **genere un fichero** de números enteros **ordenados inversamente** al anterior, es decir, debería generar en el caso del ejemplo:

7	1	9	8	2	6	5	3
---	---	---	---	---	---	---	---

 eof

```
const n=8;
var f,g: file of integer;
i,j:integer;
begin
assign(f,'ent1');
assign(g,'ent2');
reset(f);
rewrite(g);
for i:=1 to n do begin
```

end;

```
close(f);
close(g)
end.
```

7. Los números de dorsales de una carrera atlética están comprendidos entre el 1 y el 10. Los dorsales de los distintos atletas según van llegando a la meta se almacenan en un archivo de nombre 'enter2'. Se pide completar el programa para que **genere otro archivo** de números enteros 'enter3' que contenga las posiciones de llegada del dorsal 1, dorsal 2, dorsal 3, ..., dorsal 10. Por ejemplo, si el archivo de llegada de dorsales 'enter2' es:

```
10 9 8 7 6 5 1 2 3 4
eof
```

Entonces, el archivo generado 'enter3' deberá ser:

```
7 8 9 10 6 5 4 3 2 1
eof
```

```
var g,h:file of longint;
    i,j,k:longint;
begin
assign(g,'enter2');
assign(h,'enter3');
rewrite(g);
rewrite(h);
for i:=1 to 10 do
    begin read(j); write(g,j) end;
reset(g);
for i:=1 to 10 do begin
```

Se introducen los datos que generan enter2

```
end;
close(g); close(h);
(* Resto del programa... *)
```

Si enter2 fuera:

```
1 3 5 7 9 10 2 4 6 8
eof
```

 , se generara enter3:

```
1 7 2 8 3 9 4 10 5 6
eof
```

8. Completar el código de la siguiente unidad para que permita la utilización de una función recursiva factorial que calcule el factorial de un número entero.

```
UNIT EJEMPLO;
```

```
USES CRT;
FUNCTION FACTORIAL(n:Integer):real;
```

```
FUNCTION FACTORIAL;
Begin
    If n=0 then factorial:=1
```

```
    Else
```

```
End;
Begin
    .....
end.
```

9. Completar el programa siguiente para que **genere una nueva matriz** dinámica cuyos elementos tomen el valor de la suma del índice de la fila más el índice de la columna. La matriz deberá estar apuntada por un puntero *p*. El programa deberá visualizar por pantalla la matriz a la que apunta *p* en su formato de matriz:

```
type mat=array[1..4,1..4] of byte;
      pmat=^mat;
var p:pmat; i,j:byte;
begin
```

end.

Al ejecutarse, se deberá visualizar por pantalla la matriz:

```
2  3  4  5
3  4  5  6
4  5  6  7
5  6  7  8
```

10. Dada una lista simplemente enlazada de elementos completar el procedimiento *elpar* para que este **modifique una lista** también simplemente enlazada de tal manera que la lista resultante contenga solamente los elementos que ocupan una posición **impar** en la lista original. Se sabe que la lista original o está vacía o contiene un número impar de elementos. Por ejemplo, si la lista original está formada por los elementos 1 , 2 , 3 , 4 , 5 , 6 , 7 , la ejecución del procedimiento *elpar*(*v*) deberá dejar la lista con sólo los elementos 1 , 3 , 5 , 7.

```
type ptr=^elemento;
      elemento=record
          dato:integer;
          sig:ptr;
      end;
procedure elpar(v:ptr);
var p:ptr;
begin
if v<>nil then
```

end;