



Universidad
de Huelva

INTRODUCCIÓN A LA PROGRAMACIÓN - Septiembre 2002 -

*Dpto. Ing. Electrónica, Sistemas Informáticos y Automática
E.P.S. La Rábida (Universidad de Huelva)*

Apellidos :	DNI :
Nombre :	Especialidad :

NOTA:

- La **duración del examen** es de 2 horas y 30 minutos.
- Se dispone de 20 minutos antes de que comience a contar la convocatoria.
- El tiempo de resolución de dudas es de 30 minutos a partir de que comience el examen.
- A la entrega, el alumno deberá mostrar su DNI.
- Cada ejercicio debe realizarse en un folio distinto. La presentación debe ser clara, sin tachones y razonando las respuestas.
- En la cabecera de cada folio debe figurar (en el siguiente orden) el n° de ejercicio, los apellidos, nombre, DNI y especialidad.
- **Puntuación :** Ejercicio 3: 3,5 ; Ejercicio 4: 3,5

PROBLEMAS

1.- (1,5 puntos) Se desea llevar un control de los coches que pasan por determinados puntos de nuestras carreteras, para facilitar la tarea dispone de las siguientes acciones y funciones.

funcion calcular media(d,c:entero;p,m:tabla[1..30]de carácter): real;

(* dado un día d en un mes m y una cantidad c de coches que pasan por un punto p, devuelve la media de coches que lo atraviesan cada 15 minutos *)

funcion trafico_dia (d:entero;p,m: tabla[1..30]de carácter): entero

(* devuelve el n° de coches que atraviesan un punto p en un día d en un mes m *)

accion copiar(a: tabla[1..30]de carácter; var b: tabla[1..30]de carácter)

(* copia una cadena a en b *)

Diseño: a.- Una función media_totales, que calcule la media total de coches que atraviesan un punto, conocido como NIV3002, durante el mes de Junio.

b.- Una acción que guarde en alguna estructura la media de coches que pasaron por el punto conocido como NIV3002, durante los meses de Junio, Julio y Agosto. Hago uso de diseño con el fin de calcular los datos de los meses anteriores.

NOTA: Elija los parámetros de forma adecuada y razonada. Comente y razone la solución de cada uno de los apartados.

2.- (2,25 puntos) Calendario Perpetuo

Este es el llamado Calendario Perpetuo del siglo XX que nos va a permitir saber qué día de la semana fue una fecha concreto comprendida entre los años 1901 y 2000, por ejemplo vuestra fecha de nacimiento.

A) Años	B) Meses
1901 - 2000	E F M A M J J A S O N D
25 53 81	40 0 3 5 1 3 6 2 4 0 2
26 54 82	51 1 4 6 2 4 0 3 5 1 3
27 55 83	62 2 5 0 3 5 1 4 6 2 4
28 56 84	03 4 0 2 5 0 3 6 1 4 6
01 29 57 85	25 5 1 3 6 1 4 0 2 5 0
02 30 58 86	36 6 2 4 0 2 5 1 3 6 1
03 31 59 87	40 0 3 5 1 3 6 2 4 0 2
04 32 60 88	51 2 5 0 3 5 1 4 6 2 4
05 33 61 89	03 3 6 1 4 6 2 5 0 3 5
06 34 62 90	14 4 0 2 5 0 3 6 1 4 6
07 35 63 91	25 5 1 3 6 1 4 0 2 5 0
08 36 64 92	36 0 3 5 1 3 6 2 4 0 2
09 37 65 93	51 1 4 6 2 4 0 3 5 1 3
10 38 66 94	62 2 5 0 3 5 1 4 6 2 4
11 39 67 95	03 3 6 1 4 6 2 5 0 3 5
12 40 68 96	14 5 1 3 6 1 4 0 2 5 0
13 41 69 97	36 6 2 4 0 2 5 1 3 6 1
14 42 70 98	40 0 3 5 1 3 6 2 4 0 2
15 43 71 99	51 1 4 6 2 4 0 3 5 1 3
16 44 72 00	62 3 6 1 4 6 2 5 0 3 5
17 45 73	14 4 0 2 5 0 3 6 1 4 6
18 46 74	25 5 1 3 6 1 4 0 2 5 0
19 47 75	36 6 2 4 0 2 5 1 3 6 1
20 48 76	40 1 4 6 2 4 0 3 5 1 3
21 49 77	62 2 5 0 3 5 1 4 6 2 4
22 50 78	03 3 6 1 4 6 2 5 0 3 5
23 51 79	14 4 0 2 5 0 3 6 1 4 6

C) Días de la semana

Domingo	1 8 15 22 29 36	EJEMPLO: ¿Qué día de la semana fue el 4 de Julio de 1930?
Lunes	2 9 16 23 30 37	Viernes
Martes	3 10 17 24 31	Búscase el año en cuestión (1930) en la tabla A y sígase la línea horizontal a la derecha hasta llegar al número que corresponde a la columna del mes de julio de la tabla B, en este caso el dos (2). Añádase a este número (2) el guarismo del día del mes (4), con lo que resulta un total de 6. La tabla C muestra que fue un viernes el día que corresponde al número 6 hallado.
Miércoles	4 11 18 25 32	
Jueves	5 12 19 26 33	
Viernes	6 13 20 27 34	
Sábado	7 14 21 28 35	

Se pide:

- a) Dar una declaración de tipos y variables de las estructuras de datos que hay manejar para resolver el problema.
- b) Implementar una acción que pida una fecha correcta desde teclado y muestre por pantalla qué día de la semana fue la fecha introducida.

NOTA: Se evaluará el uso de las estructuras iterativas y condicionales más adecuadas para la resolución del problema.

3.- (2,75 puntos) Sea una tabla de números enteros positivos.

a) Diseñe una acción que devuelva la tabla ordenada. La cabecera de lo acción será:

accion ordena(var t: datos: n:entero);

(* dada la tabla t devuelve dicha tabla con los elementos ordenados de menor a mayor, n indica el n° de elementos de la tabla *)

(* tipo datos = tabla[1..50] de entero; *)

Ej: Supongamos que la tabla t tiene los siguientes elementos

Tabla T antes de la llamada								
20	7	1	9	22	7	9	7	14
1	2	3	4	5	6	7	8	9

Tabla T después de la llamada								
1	7	7	7	9	9	14	20	22
1	2	3	4	5	6	7	8	9

Para implementar la acción ordena el alumno obligatoriamente deberá realizar repetidas llamadas a la acción minimosiguiente (que suponemos ya implementada) y en función de los datos devueltos por ésta realizar la ordenación:

accion minimosiguiente (t: datos; n:entero; minimo: entero; var nuevominimo; var nveces: entero);

(* Entrada: t: tabla de enteros, n: n° de elementos de la tabla, minimo: mínimo anterior

Salida: nuevominimo: valor mínimo superior al mínimo antiguo.

nveces: n° de veces que aparece nuevominimo. (0 si no hay un nuevo mínimo *)

b) Implementar la acción minimosiguiente.

Básicamente dicha acción lo que calcula es el valor más pequeño de la tabla que es inmediatamente superior al indicado en el parámetro mínimo y devuelve el n° de veces que aparece dicho valor.

Ej: Supongamos que la tabla t tiene los siguientes elementos (n=13)

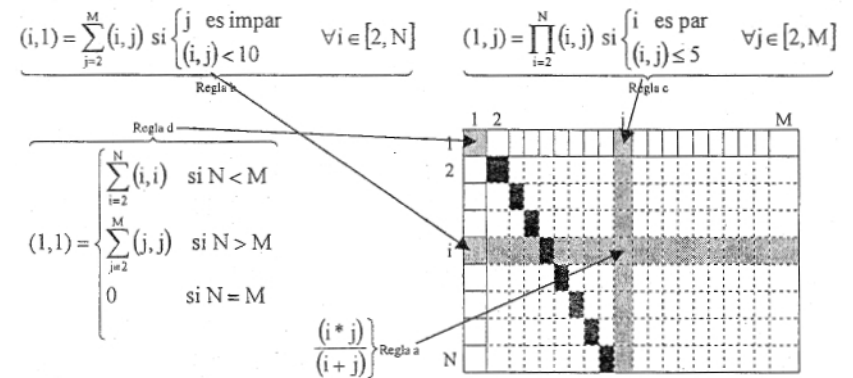
20	7	1	9	22	7	9	0	14	1	9	6	9
1	2	3	4	5	6	7	8	9	10	11	12	13

Llamada realizada
minimosiguiente (t,n,-1,nuevominimo,nveces)
minimosiguiente (t,n,7,nuevominimo,nveces)
minimosiguiente (t,n,22,nuevominimo,nveces)

Valores devueltos
nuevominimo: 0 nveces: 1
nuevominimo: 9 nveces: 4
nuevominimo: ¿? nveces: 0

4. (3,5 puntos) Realizar una acción denominada **Rellenar** que dado una tabla de reales (de tipo TTablaDatos), de tamaño NxM (N Filas y M Columnas) la devuelva rellena de datos que deben verificar las siguientes reglas:

- Para cada posición de la tabla cuya fila i esté comprendida entre 2 y N, y cuya columna j esté comprendida entre 2 y M, asignar el valor $(i*j)/(i+j)$
- Para cada posición de la tabla cuya columna sea la número 1, y cuya fila i esté comprendida entre 2 y N, asignar la suma de las columnas j comprendidas entre 2 y M, siempre y cuando j sea impar y el valor de la tabla en la posición (i, j) sea menor de 10.
- Para cada posición de la tabla cuya fila sea la número 1, y cuya columna j esté comprendida entre 2 y M, asignar el producto de las filas comprendidas entre 2 y N, siempre y cuando i sea par y el valor de la tabla almacenada en la posición (i, j) sea menor o igual a 5.
- En la posición (1,1) de la tabla se asignará la suma de la diagonal principal de la tabla, siempre y cuando el número de filas (N) sea distinto al número de columnas (M) ó 0 en caso contrario.



Un ejemplo de llamada de la acción para una tabla denominada datos, con un tamaño de 5 filas y 10 columnas sería:

Rellenar(Datos,5,10)

INTRODUCCIÓN A LA PROGRAMACIÓN

SOLUCIÓN DEL EXAMEN

- Septiembre 2002 -

Dpto. Ing. Electrónica, Sistemas Informáticos y Automática
E.P.S. La Rábida (Universidad de Huelva)

PROBLEMA 1

a.

```
funcion media totales Junio(p:tabla[1..30] de carácter): entero;  
var  
    d,suma:entero;  
    m: tabla[1..30] de caracter  
fvar  
inicio  
    suma:=0;  
    m[1]:='J'; m[2]:='U'; m[1]:='N'; m[1]:='T'; m[1]:='O';  
    para d:=1 hasta 30 hacer suma:=suma+Tráfico_dia(d,p,m);  
    retorna (suma div 30)  
función
```

b.

```
accion guardar (p:tabla[1..30] de carácter; var meses: tabla[1..3] de entero);  
var  
    i,jl,ag:entero;  
fvar  
inicio  
    j:= media_totales_Junio(p);  
    jl:= media_totales_Julio(p); (*procedimiento similar al anterior, pero con variación de días*)  
    ag:= media_totales_Agosto(p); (*procedimiento similar al primero, pero con variación de días*)  
    meses[1]:= j;  
    meses[2]:= jl;  
    meses [3]:=ag;  
faccion
```

PROBLEMA 2

a.

```
tipo  
    Annos = tabla [ 1..4,1..28] de entero;  
    Meses = tabla [ 1.. 12,1..28] de entero;  
    Dias = tabla [ 1..6,1..7] de entero;  
ftipo  
  
var  
    A: Annos;  
    B: Meses;  
    C: Dias;  
fvar
```

b.

accion Cal Per (TA:Annos; TB:Meses; TC:Dias);

var

dia:1..31;
mes: 1.. 12;
anno:1901..2000;
i,j,x:integer;
encontrado:booleano

fvar

inicio

escribir("Introduce una fecha correcta: ");
escribir("Dia: "); leer(dia);
escribir("Mes: "); leer(mes);
escribir("Anno: "); leer(anno);

encontrado:=falso;

i:=1;

mientras (i<=4) y (no encontrado) hacer

inicio

j:=1;

mientras (j<=28) y (no encontrado) hacer

si TA[i,j] = anno → encontrado := verdad

□ TA[i,j] <> anno → j: j+1;

i:=i+1;

fin;

x:= dia + TB[mes,j];

encontrado:=falso;

i:=1;

mientras (i<=6) y (no encontrado) hacer

inicio

j:=1;

mientras (j<=7) y (no encontrado) hacer

si TC[i,j] = x → encontrado := verdad

□ TC[i,j] <> x → j: j+1;

i:=i+1;

fin;

caso j de

inicio

1: escribir("DOMINGO")

2: escribir("LUNES")

3: escribir("MARTES")

4: escribir("MIERCOLES")

5: escribir("JUEVES")

6: escribir("VIERNES")

7: escribir("SABADO")

fin;

faccion;

PROBLEMA 3

a.

```
accion ordena(var t: datos; n:entero);  
var  
    aux.: datos;  
    menor,nveces,nuevomenor,i,j : entero;  
fvar  
inicio  
    para i:= 1 hasta n hacer aux[i]:=t[i];  
    j:=1;  
    menor:= -1;  
    repetir  
        minimosiguiente(aux; n; menor; nuevomenor; nveces);  
        para i:= 1 hasta nveces hacer  
            inicio  
                t[j]:=nuevomenor;  
                j: j+1;  
            fin;  
        menor:=nuevomenor;  
    hasta que nveces = 0;  
faccion
```

b.

```
accion minimosiguiente(t: datos; n:entero; minimo: entero; var nuevominimo; var nveces:  
entero);  
var  
    menor,i: entero;  
fvar  
inicio  
    nuevominimo:=minimo;  
    i:=1;  
    repetir  
        si t[i] > minimo → nuevominimo:=t[i]  
        □ t[i] <= minimo → nada  
        i:=i+1;  
    hasta que (nuevominimo > minimo) o (i > n);  
  
    nveces:=0;  
    si nuevominimo>minimo →  
        inicio  
            para i:= 1 hasta n hacer  
                inicio  
                    si (t[i]<nuevominimo) y (t[i]>minimo) →  
                        inicio  
                            nuevominimo:=t[i];  
                            nveces:=1;  
                        fin  
                    □ t[i]=nuevominimo → nveces:=nveces+1  
                    □ t[i]>nuevominimo → nada;  
                fin  
            fin  
        □ nuevominimo<=minimo → nada;  
faccion
```

PROBLEMA 4

accion rellenar(var t:TablaDatos; N,M:entero)

var

i,j:entero;

vmin: entero;

fvar

inicio

(* Regla A *)

para i:=2 hasta N hacer

para j :=2 hasta M hacer

t[i,j]:= (i*j) / (i+j);

(* Regla B, para esta regla no hace falta utilizar el operador módulo *)

para i:=2 hasta N hacer

inicio

t[i,1]:=0.0; (* Valor neutro con respecto a la suma *)

para j:=3 hasta M cada 2 hacer

si t[i,j] <10 → t[i,1]:=t[i,1]+t[i,j]

□ t[i,j] >= 10 → nada;

fin;

(* Regla C *)

para j:=2 hasta M hacer

inicio

t[1,j]:=1.0; (*Valor neutro con respecto al producto *)

para i:=2 hasta N cada 2 hacer

si t[i,j] <= 5 → t[1,j]:=t[1,j]*t[i,j]

□ t[i,j] > 5 → nada;

fin;

(*Regla D*)

t[1,1]:=0.0;

si N<>M →

inicio

si N>M → vmin:=M

□ N<M → vmin:=N;

para i:=2, hasta vmin hacer t[1,1]:= t[1,1]+t[i,j];

fin

□ N=M → nada;

faccion