



Apellidos: <i>Cognoms:</i>		Nombre: <i>Nom:</i>	
-------------------------------	--	------------------------	--

Tiempo: 1 hora

No se permiten ni libros ni apuntes

Se recomienda leer atentamente los enunciados antes de contestar.

La puntuación para preguntas con múltiples opciones es:

- Pregunta correcta: 1 punto
- Pregunta incorrecta: -0,25 puntos
- Pregunta en blanco: 0 puntos

A

Las preguntas en las que se marquen varias opciones serán consideradas incorrectas.

C1. Supongamos que deseamos ordenar parcialmente un vector de 1000 elementos (calcular, por ejemplo, los 10 números más grandes del vector). ¿Qué algoritmo de ordenación sería conveniente modificar para realizar esta ordenación?

- a.- Selección
- b.- Inserción
- c.- Quick-Sort
- d.- Intercambio o Burbuja

C1. *Suposem que desitjem ordenar parcialment un vector de 1000 elements (per exemple calcular els 10 nombres més grans del vector). Quin algorisme de ordenació seria convenient modificar per a realitzar esta operació?*

- a.- *Selecció*
- b.- *Inserció*
- c.- *Quick-Sort*
- d.- *Intercanvi o Bombolla*

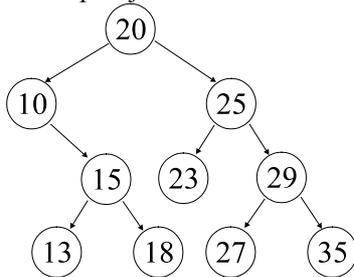
C2. Supongamos un lenguaje de programación que tiene como tipo base, el tipo Pila, pero no tiene como tipo base ni el tipo vector ni punteros. ¿Sería posible la implementación del tipo abstracto de datos Cola, con todas sus operaciones en ese lenguaje de programación?

- a.- Sí, pero la implementación del T.A.D. cola se complica.
- b.- No, ya que sólo es posible implementar los T.A.D. mediante la utilización de vectores y punteros.
- c.- Sí, pero habría que modificar las operaciones básicas de la cola
- d.- No, ya que es imposible implementar colas mediante pilas.

C2. *Suposem un llenguatge de programació que te com a tipus base, el tipus Pila, pero no te com a tipus base ni vectors ni punters. Seria possible la implementació del tipus de dades abstracte Cua, amb totes les seues operacions en eixe llenguatge de programació?*

- a.- *Sí, pero l'implementació del T.A.D. cua es complica.*
- b.- *No, ja que sols es possible implementar els T.A.D. mitjançant l'utilització de vectors i punters.*
- c.- *Sí, pero hi habria que modificar les operacions bàsiques de la cua.*
- d.- *No, ja que es imposible implementar cues mitjançant piles.*

C3. Sea el siguiente árbol binario de búsqueda. Di cuál es el recorrido prefijo correcto del árbol:



- a.- 10, 13, 15, 18, 20, 23, 25, 27, 29, 35
- b.- 20, 10, 15, 13, 18, 25, 23, 29, 27, 31
- c.- 13, 18, 15, 10, 23, 27, 35, 29, 25, 20
- d.- 20, 10, 25, 15, 23, 29, 13, 18, 27, 35

C3. *Siga el següent arbre binari de cerca. Digues quin es el recorregut prefixe correcte del arbre:*

C4. Supongamos un servidor en el que deseamos poner un servicio de impresión. Este servicio imprimirá los trabajos a medida que lleguen al servidor, de manera que si la impresora está ocupada imprimiendo un trabajo los trabajos que lleguen se guardarán en una estructura auxiliar hasta ser impresos. ¿Cuál de los siguiente T.A.D. será conveniente utilizar como estructura auxiliar?

- a.- Colas
- b.- Pilas
- c.- Colas de prioridad (montículos estáticos)
- d.- Grafos

C4. *Suposem un servidor en el que desitjem ficar un servici de impressió. Este servici imprimirà els treballs a mesura que apleguen al servidor, de maera que si la impresora està ocupada imprimint un treball, els treballs que apleguen es guardaràn en una estructura auxiliar fins ser impresos. Quin dels següents T.A.D. serà convenient utilitzar com a estructura auxiliar?*

- a.- *Cues*
- b.- *Piles*
- c.- *Cues de prioritat (mnticles)*
- d.- *Grafs*



Apellidos:
Cognoms:

Nombre:
Nom:

C5. Un árbol binario general, es posible implementarlo mediante una representación estática. Respecto de esta representación:

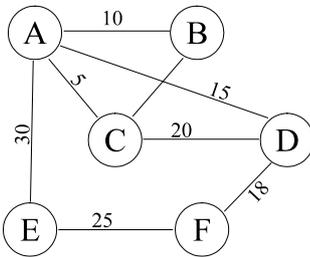
- a.- Es mejor que la dinámica porque es más fácil acceder a los hijos de un determinado nodo.
- b.- Es peor que la dinámica porque no hay que reservar dinámicamente elementos para los nuevos nodos del árbol.
- c.- Es peor que la dinámica porque el acceso al padre de un nodo es más complicado.
- d.- Sólo es aconsejable en la representación de árboles binarios llenos, por el problema de los huecos en el vector.

C5. Un arbre binari general, es posible implementarlo mitjançant una representació estàtica. Respecte de esta representació:

- a.- Es millor que la dinàmica perquè es més fàcil accedir als fills de un determinat node.
- b.- Es pitjor que la dinàmica perquè no hi ha que reservar dinàmicament elements per als nous nodes del arbre,
- c.- Es pitjor que la dinàmica perquè el accés al pare de un node es més complicat.
- d.- Sols es aconsellable en la representació de arbres binaris plens, per el problema dels 'vuits' en el vector.

C6. Dado el siguiente grafo no dirigido. ¿cuál será su representación mediante listas de adyacencia?

C6. Donat el següent graf no dirigit, ¿quina serà la seua representació mitjançant llistes de adyacencia?



¿Cuál sería el recorrido en profundidad desde el nodo 'C'?

Quin serà el recorregut en profunditat?



Apellidos:
Cognoms:

Nombre:
Nom:

C7. Sea la siguiente declaración de tipos

```
typedef Real Vector[5];
struct Reg
{
    String nombre;
    Vector * datos;
};
typedef Reg Vec[100];
```

C7. Siga la següent declaració de tipus

Si 'x' es una variable de tipo 'Vec', indicar el tipo resultante de las siguientes expresiones de acceso o si son incorrectas y por qué:

Si 'x' es una variable de tipus 'Vec', indica el tipus resultant de les següents opeacions d'accés o si son incorrectes i per qué:

```
x[2]
(* x[20].datos)[5]
x[40].nombre[1]
* x[19].datos[3]
x[30].nombre
* x[100]. * datos[3]
```

C8.- Dado el siguiente bloque de código, reescribelo primero utilizando el bucle *while* y después utilizando el bucle *for*, de manera que ambos funcionen exactamente igual que el original para cualquier caso.

```
i = 0;
cin << n;
do
{
    i = i + 1;
    cout << i;
}
while (i <= n);
```

- Con bucle *while*:

- Con bucle *for*:

C9.- Dado el siguiente vector:

```
struct Reg
{
    float f;
    int i;
}
```

Reg dato[8];

Y sabiendo que el tamaño de un real son 6 bytes, el tamaño de un entero son 4 bytes y que la variable 'dato' comienza en la posición de memoria 100, calcular en que posición de memoria está el elemento 'dato[5]'.

C9.- Siga el següent vector:

I sabent que la grandaria de un real es de 6 bytes, la grandaria de un sencer es de 4 bytes i que la variable 'dato' comença en la posició 100, calcula en quina posició de memòria està l'element 'dato[5]'.



Apellidos: Cognoms:		Nombre: Nom:	
------------------------	--	-----------------	--

P1.- Sea el siguiente programa en C++

```
#include<iostream.h>
#include<string>

string f(char c, string s)
{
    string s2, s3;
    if (s!="")
    {
        s2 = s.substr(1, s.length()-1);
        s3 = f(c,s2);
        if (s[0] == c)
            s = s3;
        else
            s = s[0] + s3;
    }
    return s;
}

int main(void)
{
    string s;

    s = "ana";
    s = f('a', s);
    cout << s;
    return 0;
}
```

Realiza una traza del programa y di cuál será el valor final de s.

length(): Método que devuelve el número de caracteres del *string*.
substr(ini, num): Método que devuelve un substring que empieza en la posición *ini* (inclusive) y contiene *num* caracteres.

P1.- Siga el següent programa en C++

Realitza la traça del programa i digues quin serà el valor final de 's'.

length(): Mètode que torna el número de caracters del *string*.
substr(ini, num): Mètode que torna un substring que comença en la posició *ini* (inclusive) i conté *num* caracters.

P2.- Supongamos la clase árbol binario que contiene en cada nodo, como información, un valor entero. Realizar un nuevo método, 'Cuentame', para la clase árbol que nos diga el número total de hojas con información negativa.

P2.- Suposem la classe arbre que conté en cada node, com a informació, un valor sencer. Realitzar un nou mètode, 'Cuentame', per a la classe arbre que ens diga el numero total de fulles amb informació negativa.

P3. Añadir un nuevo método a la clase Lista (con punto de interés) que invierta el orden de los elementos de la lista, suponiendo una implementación dinámica con enlace simple, sin nodo cabeza y no circular de la misma. El punto de interés quedará situado al inicio de la nueva lista.

P3.- Afegir un nou mètode a la classe Llista (amb punt de interes) que invertisca l'orde dels elements de la llista, suposant una implementació dinàmica amb enllaç simple, sense node cap i no circular. El punt d'interés es quedarà situat al inici de la nova llista.