
 71901020		Fundamentos de Programación		901
		MULTIPLES GRADOS		
	Febrero - 2014 Original	Duración: 120 min.	EXAMEN: Tipo A Desarrollo	Nacional 1º Cuatrimestre
[{}>{}] Material: Ninguno				Hoja 1 de 3

FUNDAMENTOS DE PROGRAMACION	MODELO 1	CONV. ORDINARIA 13/14
MATERIAL AUTORIZADO: NINGUNO		DURACIÓN: 2 HORAS
INSTRUCCIONES:	1) El test debe ser contestado en la hoja de marcas. Sólo una de las cuatro respuestas posibles de cada pregunta es correcta. 2) El test es eliminatorio y aporta un 30% de la nota final. Son necesarias 8 preguntas correctas (6 con las prácticas aprobadas) para que se corrija el ejercicio. 3) Cada respuesta correcta 1 pts. Respuesta incorrecta, doble o en blanco: 0 pts. 4) Puede quedarse, si lo desea, con esta hoja de examen.	

CUESTIONES DE TEST

1. El siguiente fragmento de programa en C±:

```
void algo (uno dos)
```

- A.- Es la llamada a un procedimiento con un argumento
- B.- Es la cabecera de un procedimiento con dos argumentos
- C.- Es la cabecera de un procedimiento con un argumento
- D.- Es la llamada a un procedimiento con dos argumentos

2. En el manual de estilo del lenguaje C± está prohibido:

- A.- Utilizar punteros apuntando a formaciones
- B.- Utilizar formaciones de punteros
- C.- Utilizar los punteros como formaciones
- D.- Utilizar formaciones sin emplear punteros

3. En el siguiente fragmento de programa en C±:

```
extern uno dos
```

- A.- La variable dos se tendrá que acabar de definir en el módulo de implementación
- B.- El tipo uno se tendrá que acabar de definir en el módulo de implementación
- C.- Forma parte de un módulo de implementación
- D.- Forma parte del módulo principal

4. En el siguiente fragmento de programa en C±:


```
while ((Uno < 10)&&(Dos != 'z'))
```

- A.- Dos puede ser una variable entera
- B.- Dos puede ser una constante entera
- C.- Uno puede ser una constante carácter
- D.- Uno puede ser una variable entera

5. La ejecución del siguiente código:

- A.- Imprime 0 0 1 2 1 0 1 2 2 0 1 2
- B.- Imprime 0 0 0 1 1 1 2 2
- C.- Imprime 0 0 0 1 1 2
- D.- Produce un error

```
for (int i=0;i<3;i++) {
    printf("%d ", i);
    j = i;
    while (j<3) {
        printf("%d ", i);
        j++;
    }
}
```

 71901020		Fundamentos de Programación		901
		MULTIPLES GRADOS		
Febrero - 2014 Original	Duración: 120 min.	EXAMEN: Tipo A Desarrollo	Nacional 1° Cuatrimestre	
[{}>{}] Material: Ninguno			Hoja 2 de 3	

6. En C±:

- A.- no existen ficheros con extensión .o
- B.- las extensiones .h, .cpp y .o corresponden a ficheros de implementación, interfaz y objeto, respectivamente
- C.- las extensiones .h, .cpp y .o corresponden a ficheros de interfaz, implementación y objeto, respectivamente
- D.- no existen ficheros con extensiones .o ni .h

7. La ordenación de un vector por inserción directa:

- A.- requiere dos bucles
- B.- sólo requiere un bucle
- C.- requiere tres bucles
- D.- no requiere ningún bucle

8. ¿Cuál de los siguientes identificadores NO es válido en C+/- ?

- A.- var_8_iable
- B.- Error_val
- C.- variable_8
- D.- 8variable

9. Dado el siguiente programa:

¿Cuál es el valor final de numero?

- A.- 10
- B.- 40
- C.- 1234
- D.- 1

```
int sum=0;
int r;

int Revolver(int num){
    if (num > 0){
        r=num%10;
        sum=sum+10;
        return(Revolver(num/10));
    }
    else {return sum;}}

int main(){
    int numero = 1234;
    numero = Revolver(numero);
}
```

10. El paso como argumento de un puntero equivale a pasar la variable apuntada:

- A.- siempre por valor
- B.- siempre es por valor si es de tipo entero
- C.- en algunos casos por valor y en otros por referencia según el tipo
- D.- siempre por referencia

 71901020		Fundamentos de Programación		901
		MULTIPLES GRADOS		
Febrero - 2014 Original	Duración: 120 min.	EXAMEN: Tipo A Desarrollo	Nacional 1° Cuatrimestre	
[{}>{}] Material: Ninguno			Hoja 3 de 3	

EJERCICIO DE PROGRAMACIÓN

Realizar el tipo abstracto de datos **Boleto** para guardar los 15 números de una apuesta. La operación **NuevoBoleto** lee de teclado los 15 números seleccionados por el jugador y los guarda. La operación **MostrarBoleto** tiene un argumento que permite mostrar en pantalla desde el primero al último número o bien al contrario. La operación **OrdenaBoleto** ordena los números del boleto.