FUNDAMENTOS DE PROGRAMACION	MODELO 3	*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 pto. Respuesta incorrecta, doble o en blanco: 0 pto.	
	4) Puede quedarse, si lo desea, con esta hoja de examen.	

CUESTIONES DE TEST

1. La notación BNF especifica:

A.- la sintaxis de un lenguaje
B.- el orden de complejidad de un algoritmo
C.- programas funcionales
D- formatos numéricos binarios

2. La ejecución del siguiente código:

```
for (int i=1;i<3;i++) {
  printf("%d ", i);
  j = i;
  while (j<2) {
    printf("%d ", i);
    j=j-1;
  }
}
```

A.- No termina nunca
B.- Imprime 1 1 0 2
C.- Imprime 1 2
D.- Produce un error de compilación

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

```
#include <Uno.h>
#include "Dos.h"
```

Se indica al compilador que los ficheros del módulo…

A.- Uno se deben buscar donde está instalada la herramienta de compilación
B.- Uno y Dos se deben buscar donde está instalada la herramienta de compilación
C.- Uno se debe buscar donde reside el código fuente de la aplicación
D.- Uno y Dos se deben buscar donde reside el código fuente de la aplicación

4. Si en C+/- aparece la declaración constante:

```
const char ValorMaximo[]="32.0";
```

Se trata de una constante de tipo:

A.- Cadena de caracteres
B.- Entero
C.- Real
D.- Coma flotante

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

```
case algo :
```

A.- algo debe ser una expresión constante
B.- algo puede ser una variable de cualquier tipo
C.- algo debe ser una variable de tipo entero
D.- algo puede ser cualquier expresión

6. La ejecución del siguiente código:

```
try {
  if ((5+2*3%3)==0) {throw 0;}
  else {throw 1;}
  printf("%d ", 2);
} catch (int e) {
  printf("%d ", e);
}
```

A.- Imprime 1
B.- Imprime 0
C.- Imprime 1 2
D.- Imprime 0 2

7. ¿Cuál sería la función que emplearía para calcular la suma de todas las cifras de un número entero positivo dado?

A.-

```
int Suma  (int valor){
 if (valor == 0) { return 0;}
 else {
   return((valor %10)+Suma(valor / 10));
 }
}
```

B.-

```
int Suma  (int valor){
 if (valor == 0) { return 1;}
 else {
   return((valor %10)+Suma(valor / 10));
 }
}
```

C.-

```
int Suma (int valor){
  int total =0; int i = valor;

  while  (i > 9){
     total = i % 10 + total; i = i / 10;
  }
    return (total);
}
```

D.-

```
int Suma (int valor){
  int total =0; int i = valor;

  while  (i > 10){
     total = i % 10 + total; i = i / 10;
  }
    return (total);
}
```

8. ¿Cuál es la operación que realiza el subprograma Convertir?

A.- Pasa sólo las mayúsculas a minúsculas
B.- Pasa sólo las minúsculas a mayúsculas
C.- No convierte nada
D.- Convierte las mayúsculas a minúsculas y las minúsculas a mayúsculas

```
typedef char TipoVector[6];

void Convertir1 (TipoVector V){
  for (int i=0; i < 5; i++){

    if (!(islower(V[i]))){
      V[i] = tolower(V[i]);}
    if (isupper(V[i])){
      V[i] = toupper(V[i]);}
  }}
```

9. En el fichero de interfaz la declaración de los subprogramas en C+/- incluye:

A.- el tipo, el nombre y los argumentos.
B.- únicamente el nombre si es un procedimiento
C.- si es una función sólo el tipo devuelto y el nombre
D.- sólo el nombre de la función o el procedimiento

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

```
void  Uno :: Dos ()
```

A.- Uno es un tipo abstracto de dato

B.- Dos es una función sin argumentos

C.- Uno es una función sin argumentos

D.- Dos es un tipo abstracto de datos

EJERCICIO DE PROGRAMACIÓN

Realizar un tipo abstracto de datos Binario que permita manejar números binarios del 0 al 65536 (2^16). Las operaciones que se deben resolver son PonerCero, SumarBinarios, y ConvertirEnteroaBinario. La operación PonerCero recibe un número binario y lo inicia a cero. La operación SumarBinarios recibe dos números binarios y devuelve el número binario resultado de la suma (si se supera el valor máximo definido en el TAD entonces devuelve el binario cero). La operación ConvertirEnteroaBinario recibe un valor de tipo entero dentro del rango y lo convierte a su número binario.