

| | | | | | |
|--|------------------------|--|-----------------------------|---|-------------------------|
| Familia Profesional Informática y Telecomunicaciones | | Nombre del Ciclo Formativo Título de Técnico Superior en Desarrollo de Aplicaciones Web | | | |
| Centro Educativo IES Campanillas (sede CITIC) | | Módulo Profesional Desarrollo Web en Entorno Servidor Código: 0613 N.º de créditos ECTS: 12 | | Profesor Luis José Sánchez González | |
| Curso lectivo 2015 / 2016 | Grupo 2º DAW | Tipo de documento Examen | Trimestre Primero | Fecha 21 de octubre de 2015 | Turno Primero |

INSTRUCCIONES

- ➔ El alumno debe entregar una carpeta con las soluciones al examen cuyo nombre debe estar formado por el número de lista seguido de las iniciales. Por ejemplo, Facundo Romuedo Piladro que es el número 8 de la lista entregaría una carpeta con nombre **Ex08frp**.
- ➔ Los ficheros o carpetas correspondientes a las soluciones se deben nombrar igual que la carpeta junto con el número del ejercicio, por ejemplo **Ex08frp1.php**, **Ex08frp2.php**, etc. en caso de ser ficheros o **Ex08frp1**, **Ex08frp2**, etc. en caso de ser carpetas.
- ➔ En los comentarios de cada programa **se debe indicar el nombre completo**, la fecha y - si procede - el turno.
- ➔ Únicamente se necesita entregar el código fuente en php junto con las imágenes y las hojas de estilo en caso de que las haya. No se deben entregar los ficheros o carpetas con información del proyecto (por ej. la carpeta nbproject).

EJERCICIOS

1. Escribe un programa que pida 6 números uno detrás de otro (no se pueden pedir los 6 en la misma página) y que, a continuación, muestre el máximo, el mínimo y el número de primos (solo la cantidad).

2. Realiza un programa que pida 10 números por teclado y que los almacene en un *array*. A continuación se debe mostrar el contenido de ese *array* junto al índice (0 – 9). Seguidamente el programa debe colocar de forma alterna y en orden los pares y los impares: primero par, luego impar, luego par, luego impar... Cuando se acaben los pares o los impares, se completará con los números que queden.

Ejemplo:

Array original:

| | | | | | | | | | |
|----|---|---|---|----|---|---|----|----|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 20 | 5 | 7 | 4 | 37 | 9 | 2 | 17 | 11 | 6 |

Array resultante:

| | | | | | | | | | |
|----|---|---|---|---|----|---|---|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 20 | 5 | 4 | 7 | 2 | 37 | 6 | 9 | 17 | 11 |

3. En ajedrez, el valor de las piezas se mide en peones según la siguiente tabla:

| Pieza | Dama | Torre | Alfil | Caballo | Peón |
|-------|----------|----------|----------|----------|--------|
| Valor | 9 peones | 5 peones | 3 peones | 2 peones | 1 peón |

Realiza un programa que vaya generando al azar las piezas que capturan dos jugadores durante una partida. Un jugador se rinde cuando el contrario captura el equivalente a 9 peones (o más).

Ejemplo:

| |
|--|
| Fichas capturadas: Alfil negro (3 peones) Caballo blanco (2 peones) Peón blanco (1 peones) Torre negro (5 peones) Alfil negro (3 peones) Las negras se rinden, han perdido el equivalente a 11 peones. |
|--|

Hay que tener en cuenta que cada jugador tiene la posibilidad de capturar algunas de las siguientes piezas (no más): 1 dama, 2 torres, 2 alfiles, 2 caballos y 8 peones.

El valor de cada pieza se debe almacenar en un *array* asociativo.

4. Crea un array bidimensional de 10 filas por 9 columnas relleno con números aleatorios entre 1 y 1000 (ambos incluidos). Los números se pueden repetir. Se deben mostrar todos los números bien alineados en filas y columnas. Muestra el máximo de entre los números capicúa en azul y los números adyacentes en verde. Si el máximo capicúa se repite, se puede colorear uno cualquiera de ellos o todos, como al alumno le resulte más fácil.