

Examen Pendientes 1º DAM – Parte 1

• Módulo: Programación

• Alumno:

1. (2,5 ptos.) Realiza un programa que compruebe si dos números son amigos.

"Dos números amigos son dos números enteros positivos tales que la suma de los divisores propios (divisores excepto él mismo) de uno es igual al otro número y viceversa."

Un ejemplo es el par de números 220 y 284, ya que:

- los divisores propios de 220 son 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110, que suman 284.
- los divisores propios de 284 son 1, 2, 4, 71 y 142, que suman 220.

NOTAS:

- Los números serán introducidos por teclado, y deben de cumplir las restricciones indicadas, de lo contrario serán solicitados nuevamente hasta que sean válidos.
- El número de intentos de introducción de los datos válidos será de 3. Si se supera este límite, el programa finalizará con un mensaje de error.

2. (2,5 ptos.) Realiza un programa que determine el nivel de seguridad de una contraseña introducida por teclado. Para ello considera los siguientes niveles de seguridad:

- **Bajo:** la contraseña debe tener un mínimo de 5 caracteres.
- **Mediano:** la contraseña debe tener un mínimo de 6 caracteres, y debe incluir números, letras en mayúsculas y minúsculas.
- **Alto:** cada contraseña debe tener un mínimo de 8 caracteres, y debe incluir números y letras en mayúsculas y minúsculas, y al menos un carácter especial que no sea ni una letra ni un número.

NOTA:

- Las contraseñas que no lleguen a 5 caracteres no serán permitidas, en cuyo caso el programa finalizará con un mensaje informativo.

3. (2,5 ptos.) Realiza un programa que simule una carrera entre 2 camellos, con la misma probabilidad de ganar. Para ello ten en cuenta las siguientes indicaciones:

- El trayecto a recorrer son 100 metros, hasta llegar a la meta.
- Cada vez que avanza un camello, lo hace 1 metro.
- Para no aplicar programación multihilo, se va a determinar qué camello "sale" antes, aleatoriamente.

NOTA:

- Para simular el avance de un camello elige un carácter y píntalo en la consola cada vez que recorra 1 metro.

4. (2,5 ptos.) Diseña la clase *Persona* teniendo en cuenta las siguientes indicaciones:

- Atributos:
 - *idPersona*: identificador único de la persona. Numérico y correlativo.
 - *nombre*: cadena con el nombre de la persona.
 - *sexo*: carácter que indica el sexo de la persona. Valores H (hombre) o M (mujer).
 - *padre*: de tipo *Persona*.
 - *madre*: de tipo *Persona*.
- Constructor/es:

Se podrá crear una persona proporcionando su nombre, o además del nombre, proporcionando su padre y su madre. En este caso, si alguno de ellos (padre o madre) se omite se lanzará una excepción.
- Métodos:
 - *toString()*: devuelve la información de la persona con el siguiente formato:

Persona (ID: <idPersona> Nombre: <nombre> Sexo: <sexo>
Padre: <nombre padre> Madre: <nombre madre>
 - *sonHermanos(Persona otra)*: determina si la persona actual y otra persona recibida como parámetro son hermanos. Dos personas son hermanos si tienen el mismo padre y la misma madre.
 - *huerfano()*: método que deja a la persona actual sin padre y sin madre.
 - *familiaHomoparental()*: método que determina si la familia de la persona está formada por progenitores del mismo sexo, es decir, dos padres o dos madres.