|  |  |  |
| --- | --- | --- |
|  | **PROGRAMACIÓN**  **1º CFGS DAM**  **Examen – Segunda Evaluación** |  |

Para todo el examen, los programas tienen que estar **modulados**, en otro caso el ejercicio valdrá cero. Los programas se realizarán en java con NetBeans.

**Forma de entrega** se comprimen los tres ejercicios en un zip y se suben a la plataforma de educamos, el profesor también pasará por los ordenadores para copiarlo en pendrive, en este caso se tienen los tres ejercicios metidos en una carpeta sin comprimir con vuestro nombre y apellidos.

Ejercicios.

1.- Realizar un programa modular que guarde los cien primeros números en un vector, después pedirá un número del 0 al 9 por teclado, y mostrará por pantalla todos los números que no tengan el dígito introducido por teclado. (2 puntos).

2.- Vamos a simular la quiniela de futbol, pero solamente con 15 resultados.

Partidos: Un partido puede ser ganado, empatado o perdido. Se jugarán 15 partidos, cada uno de ellos servirá para un resultado en la quiniela. La simulación del resultado del partido será aleatoria, deben de guardarse el resultado de todos los partidos, de cada jornada.

Quiniela: se crean dos quinielas uno para el ordenador y otro para nosotros donde iremos introduciendo nuestros resultados por teclado, nuestra quiniela servirá para las tres jornadas, el ordenador lo realizará aleatoriamente, una por jornada, la quiniela del ordenador puede coincidir como máximo en 5 resultados con la del humano, se tiene que ir controlando según se crea (no vale generar una nueva).

En programa principal debe implementar la simulación de las jornadas y la simulación de las quinielas y quien ha ganado en cada una de las jornadas. Se dará por ganador el que acierte más resultados, si se queda empate también se comunicará. Terminará cuando alguien gane 5 veces.(4 puntos)

3.- Tenemos una matriz de 7x7 que se rellena con números aleatorios del 1 al 9, el ordenador elegirá al azar una posición de la matriz y revisará todos sus adyacentes, si la suma de todos los adyacentes es superior a la elegida al azar \*3, deberá de poner a uno todas las posiciones de la matriz, excepto las adyacentes. Debe mostrar la matriz antes y después de cambiar. Para demostrar que ha funcionado. (4 puntos)