

INSTRUMENTO DE EVALUACIÓN

PROGRAMACIÓN		
Curso y grupo: 1º DAM	Módulo:	0373
Fecha de entrega IE: 15/01/2021	UT 6	Identificar las características de las estructuras estáticas y dinámicas de almacenamiento y utilizarlas en aplicaciones.
	I.E.	6.1.

Resultados de aprendizaje:

RA6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

Metodología

Explicación teórica y ejercicios prácticos.

Equipos y materiales:

Equipos informáticos (IDE Eclipse) apuntes de clase, manuales e Internet.

Temporalización:

16 horas.

Criterios de evaluación:

- 6.1. Conoce la importancia y utilidad de las estructuras de almacenamiento en memoria.
- 6.2. Realiza operaciones avanzadas con cadenas de caracteres.
- 6.3. Utiliza arrays unidimensionales y multidimensionales para la resolución de problemas.
- 6.4. Comprende y sabe aplicar los algoritmos de ordenación.

Actividades

1. Dado un vector de 5 enteros actualizar cada posición de dicho vector con un número leído. (0,1 puntos)
2. Obtenga la sumatoria y la media de los elementos de un array. (0,1 puntos)
3. Leer un array de 10 elementos e intercambiar los elementos situados en posiciones consecutivas (1 y 2, 3 y 4, ... , 9 y 10). (0,1 puntos)
4. Leer un array de 10 elementos enteros y mostrar el valor del elemento mayor, menor y la media aritmética. (0,1 puntos)
5. Realice un programa para leer un array ingresando solamente números múltiplos de 3 que ingresan por teclado. (0,1 puntos)
6. Leer un array de 10 elementos y efectuar una búsqueda de un elemento, mostrando la posición en la que se encuentra dicho elemento. (0,1 puntos)
7. Leer un array y efectuar una búsqueda dicotómica. (0,1 puntos)
8. Elabore un programa que permita introducir un arreglo de 25 elementos de tipo entero. Luego pedir al usuario que introduzca un número. el programa mostrara el número de veces que se repite dicho valor en el arreglo. (0,1 puntos)
9. Elabore un programa que permita introducir 20 elementos de tipo entero en un arreglo, el programa mostrara impreso el arreglo en orden inverso. (0,1 puntos)
10. Elabore un programa que permita encontrar el primer y segundo mayor de un array de 15 elementos. (0,1 puntos)
11. Realiza un programa que cree 1000 números aleatorios y muestre los 10 mayores. (0,1 puntos)
12. Realiza un programa que cree un vector de 100 posiciones con números aleatorios entre 10 y 80. Una vez creado el vector el programa deberá mostrar el mayor, el menor, el valor que más se repite y la media. (0,1 puntos)
13. Dado dos vectores A y B de 15 elementos cada uno, obtener un vector C donde la posición i se almacene la suma de $A[i]+B[i]$ y mostrar el mayor de los $C[i]$. (0,1 puntos)
14. Dado una secuencia de número leídos y almacenados en un vector A y un número leído determinar si dicho número se encuentra o no en el vector. (0,1 puntos)
15. Leer una secuencia de 20 números y almacenar en un vector sus factoriales. (0,1 puntos)
16. Dado dos matrices A y B obtener la suma. (0,1 puntos)
17. Dado una matriz determinar la posición (i, j) del mayor y menor. (0,1 puntos)
18. Leer una secuencia de 20 números almacenarlos en un vector $A[1..20]$ y mostrar la suma de los elementos que ocupan posiciones pares y el mayor de los que ocupan posiciones impares. (0,1 puntos)

19. Dado un vector de números determina aquellos que sea primos. (0,1 puntos)
20. Crear un programa llamado paresImpares que cree un array de 100 números aleatorios del 1 al 1000. Una vez creado, mostrar el contenido y después organizarlo de forma que estén juntos los elementos pares y los impares en dos vectores distintos. (0,1 puntos)
21. Dadas dos matrices A y B intercambiar los mínimos de A con los máximos de B. (0,1 puntos)
22. Dada una matriz cuadrada invertir su diagonal principal. (0,1 puntos)
23. Realiza un programa que a partir de dos arrays determine si uno está contenido en el otro. (0,1 puntos)
24. Realiza un programa que cuente el número de vocales que hay en una cadena de caracteres. (0,1 puntos)
25. Realiza un programa que realice una sustitución de una palabra por otra dentro de una cadena que tenga las palabras separadas por un carácter blanco. (0,1 puntos)
26. Leer una cadena y encriptarla empleando el método Cesar, que consiste en desplazar cada carácter 3 posiciones. (0,1 puntos)
27. Dado un String en la que solo incluimos caracteres alfabéticos, realizar las siguientes operaciones sobre ella: (0,25 puntos)
 - a. Mostrar la cadena con los caracteres invertidos.
 - b. Mostrar por consola el número de veces que aparecen las letras a, o y e en la String introducida. Además, si el número de veces que se repite la a es superior a 10 debe aparecer el mensaje "Exceso de a", si el número de veces que se repite la o es superior a 5 debe mostrarse "Exceso de o" y si se repite más de 3 veces la letra e debe mostrarse "Exceso de e".
 - c. Realizar la búsqueda de una palabra que introducimos por teclado dentro del String original.
28. Realizar un programa que lea una cadena de caracteres e indique si la cadena es un palíndromo. Un palíndromo es una cadena que se lee de igual forma de derecha a izquierda y de izquierda a derecha. (0,25 puntos)
29. Verificar si una cadena de texto almacenada en la String nif, es un NIF correcto o no. Si lo es, se mostrará por consola su parte numérica; si no lo es se mostrará el mensaje "NIF no valido". Se tendrá en cuenta lo siguiente: (0,25 puntos)
 - Suponer que los NIFs tienen 8 dígitos y, a continuación, una letra (no importa que sea mayúscula o minúscula).
 - PISTAS: dos condiciones que debe cumplir el NIF: tener 9 caracteres y que el último sea una letra. Comprobado esto, verificar que el resto de caracteres son dígitos.
30. Tenemos una cadena 'notas' con los nombres y las notas de 5 de los alumnos de clase. El contenido de la cadena es el siguiente:
"Juan Carlos \n 8.5 \n Andrés \n 4.9 \n Pedro \n 3.8 \n Juan \n 6.3"
El formato es "nombre \n nota \n..."
Realiza un programa que muestre por pantalla por cada alumno lo siguiente:
El alumno X ha sacado la nota Y.

Utiliza para realizar el programa la clase StringTokenizer. (0,25 puntos)

31. Modifica el código del ejercicio anterior para que las notas se almacenen en un vector de datos double. (0,25 puntos)
32. Realiza un método esCapicua que tome como parámetro un entero y devuelva true si el número es capicúa y false en caso contrario. (0,25 puntos)
33. Diseñar un programa que permita mover una ficha sobre un tablero de 8x8. En las celdillas de dicho tablero tenemos varias minas. El juego finaliza cuando el usuario topa con una de dichas minas. (0,5 puntos)
34. Diseñar un programa que muestre un cuadrado latino de N x N. (0,5 puntos)
35. Crear una agenda electrónica mediante un array de objetos Contacto(nombre, dirección, teléfono), permitiendo la introducción, eliminación, listado y modificación de los elementos. Realizar un menú para controlar las funcionalidades. (1 punto)
36. Crear un programa que realice una sopa de letras. La sopa de letras tendrá un tamaño de matriz 15x15. El programa pedirá 10 palabras, las cuales las irá escondiendo de forma aleatoria por la matriz (obviamente las palabras siempre tendrán 15 letras o menos). Una vez escondidas las palabras rellenará las demás casillas de la matriz con letras de forma aleatoria. Sólo se utilizarán mayúsculas. Si el usuario introduce palabras en minúsculas se transformarán a mayúsculas. (1 punto)
37. Una matriz tiene un punto de silla en una de sus componentes si es el mayor valor de su columna y el menor de su fila. Diseñar un algoritmo que recibiendo una matriz cuadrada de NxN, escriba en la pantalla las coordenadas de todos sus puntos silla. (0,5 puntos)
38. Realiza una clase con un método factorial que utilizando la recursividad genere el factorial de un número dado. (0,25 puntos)
39. Realiza un programa con un método recursivo que muestre por pantalla la serie de Fibonacci. (0,25 puntos)
40. Realiza un programa que utilizando recursividad muestre por pantalla la siguiente pirámide: (0,5 puntos)

$$\begin{array}{ccccccc} & & & & 1 & & \\ & & & & 1 & 1 & \\ & & & 1 & 2 & 1 & \\ & & 1 & 3 & 3 & 1 & \\ 1 & 4 & 6 & 4 & 1 & & \end{array}$$

41. Crear un programa que permita generar una matriz “caracol”. El programa permitirá generar un entero (que será el orden de la matriz) y devolverá la matriz “caracol” de dicho orden. Crear dos métodos, uno que sirva para rellenar la matriz caracol y otro para mostrarla. A continuación, se muestra un ejemplo de matrices caracol de orden 2, 3, 4 y 5 respectivamente. (0.5 puntos)

[illegible]

42. Tenemos una matriz de [5][5] de la cual queremos ordenar una fila concreta que vamos a especificar por teclado. A continuación, nos mostrará la matriz ordenada. (0,5 puntos)
- d. Utilizar el método de la burbuja para realizar la ordenación.
 - e. Ahora queremos ordenar la diagonal principal de la matriz. Realizar la ordenación por inserción directa.
43. Dada una matriz $N \times N$ y un vector de tamaño N , determina si existe alguna columna de la matriz que sea igual al vector, y en caso de que exista, indicar que número de columna es. Ordenar en orden ascendente dicha columna en la matriz dada (Utilizar un menú para elegir que método de ordenación se utiliza) y mostrar dicha matriz con la columna ya ordenada. (0,5 puntos)

Para la realización de los ejercicios se tendrá en cuenta las siguientes implementaciones:

- ***El control de excepciones aplicado en el código.***
- ***El código debe estar documentado utilizando las anotaciones correctas.***
- ***Los comentarios en el código.***
- ***La mejor utilización posible de las características de las estructuras de almacenamiento estáticas tanto unidimensionales como multidimensionales, las cadenas de caracteres, las clases String y StringBuffer, los algoritmos de ordenamiento y el uso de la recursividad.***

Una vez realizada la tarea se realizará la exportación de la carpeta del proyecto de Eclipse (el nombre del proyecto será Torres_Molina_Manuel_PROGUT6_I.E.6.1, donde vendrán todas las actividades desarrolladas en Java. El envío se realizará a través de la plataforma de la forma establecida para ello.