

Trabajo Práctico Nº 7

Variables multidimensionales: Vectores y Matrices

Para la resolución de cada uno de los siguientes ejercicios se solicita que diseñe y codifique un algoritmo que resuelva el problema usando la herramienta PSeInt.

El programa debe funcionar correctamente y producir una salida consistente para distintos grupos de datos de prueba según corresponda.

Es obligatorio el uso de subprogramas para resolver los ejercicios.

Ejercicios con vectores:

1. Crear un array que contenga 100 notas con valores entre 0 y 20 generadas aleatoriamente mediante el uso de la función `azar()` o `aleatorio()` de PSeInt. Luego, de acuerdo a las notas contenidas, el programa debe indicar cuántos estudiantes son:
 - Deficientes 0-5
 - Regulares 6-10
 - Buenos 11-15
 - Excelentes 16-20
2. Realizar un programa que lea N valores ingresados por el usuario. A continuación, se debe buscar un número indicado por el usuario dentro del arreglo. El programa debe indicar la posición donde se encuentra el valor. Si el número se encuentra repetido dentro del arreglo se deben imprimir todas las posiciones donde se encuentra ese valor. Finalmente, en caso que el número a buscar no se encuentre dentro del arreglo se debe mostrar un mensaje al usuario.
3. En un juego se lanza un dado 5 veces y se gana cuando la suma de puntos obtenidos en los dos primeros lanzamientos es igual a la obtenida en los restantes tres. Por ejemplo:

```
>>juego()  
dado = [3, 2, 1, 3, 1]  
sumas = [5, 5]  
respuesta=1
```

```
>>juego()  
dado = [2, 4, 5, 2, 1]  
sumas = [6, 8]  
respuesta=0
```

Escriba un programa que simule los cinco lanzamientos del dado y retorne dos posibles valores: **1**, si se trata de un jugada ganadora y **0** en caso contrario. Además, el programa debe permitir realizar **N** jugadas diferentes y mostrar por pantalla cada una de las jugadas (valores obtenidos por los dados, sumas y respuesta) y el porcentaje de jugadas ganadoras.

Ejercicios de manejo de cadenas de caracteres

4. Escribir un programa que procese una secuencia de caracteres ingresada por teclado y terminada en punto (leídos de a uno por vez), y luego codifique la palabra o frase ingresada de la siguiente manera: cada vocal se reemplaza por el carácter que se indica en la tabla y el resto de los caracteres (incluyendo a las vocales acentuadas) se mantienen sin cambios.

a	e	i	o	u
@	#	\$	%	*

Realice un subprograma que reciba una vocal y retorne la codificación correspondiente. Utilice la estructura “según” para la transformación. Además se debe calcular cuál es la consonante dentro de la frase que se repite más veces.

Por ejemplo, si el usuario ingresa: Ayer, lunes, salimos a las once y 10.

La salida del programa debería ser:

@y#r, l*n#s, s@l\$m%\$ @ l@s %nc# y 10.

La letra s es la más repetida y aparece 4 veces

Nota: investigue el uso de la función *concatenar* de PSeInt para armar la palabra/frase.

5. Un oficial de correos decide optimizar el trabajo de su oficina cortando todas las palabras de más de cinco letras a sólo cinco letras (e indicando que una palabra fue cortada con el agregado de una arroba). El oficial también elimina todos los espacios en blanco de más. Por ejemplo, el siguiente texto:

Llego mañana alrededor del mediodía

se transcribe como:

Llego mañan@ alred@ del medio@

Además, los puntos se reemplazan por la palabra especial STOP, y el punto final (que puede faltar en el texto original) se indica como STOPSTOP.

Por ejemplo, el siguiente texto:

Llego mañana alrededor del mediodía. Voy a almorzar

Se lo transcribe como:

Llego mañan@ alred@ del medio@ STOP Voy a almor@ STOPSTOP

Para enviar un telegrama se debe abonar un valor por las palabras cortas y otro valor por las palabras largas (las que deben ser cortadas).

Escriba un programa que automatice el trabajo del oficial de correos. Para ello se debe ingresar el mensaje a transmitir, el costo de cada palabra corta, y el costo de cada palabra larga. El programa debe mostrar como salida el texto del telegrama y el costo que se debe abonar por el mismo.

6. Disponemos de un array unidimensional de 20 elementos de tipo carácter. Se pide desarrollar un programa que:
- Pida una frase y luego copie carácter a carácter dentro del arreglo. Ayuda: utilizar la función *subcadena*.

- b. Una vez completado lo anterior, pedirle al usuario que ingrese un carácter cualquiera y una posición. El programa debe intentar ingresar el carácter en la posición indicada si es que hay lugar (es decir si está vacío o en blanco). De ser posible debe listar el vector con el carácter ingresado, de lo contrario debe darle un mensaje al usuario.
7. Tomando en cuenta el ejercicio anterior, mejore el mecanismo de inserción del carácter, facilitando un potencial reordenamiento del vector. Digamos que se pide ingresar el carácter en la posición X y la misma está ocupada, entonces de existir un espacio en cualquier posición X-n o X+n, desplazar los caracteres hacia la izquierda o hacia la derecha para poder ingresar el carácter en cuestión en el lugar deseado. El procedimiento de reordenamiento debe ubicar el espacio más cercano y desplazar los caracteres hasta alcanzar esa posición.

Ejemplo, suponiendo la siguiente frase y los subíndices del vector:

h	o	l	a		m	u	n	d	o		c	r	u	e	l	!			
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Si se desea ingresar el carácter “%” en en la posición 8, entonces el resultado con desplazamiento sería:

h	o	l	a		m	u	n	%	d	o	c	r	u	e	l	!			
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Notar que el desplazamiento se hizo hacia la izquierda porque el espacio de la posición 10 estaba más cerca de la posición 8 que el espacio de la posición 4. El desplazamiento se realizó hasta la posición donde se encontró el espacio más cercano.

Ejercicios con matrices:

8. Escribir un programa que dado un menú de opciones permita al usuario:
1. Crear una matriz de orden N, siendo N un número impar, con números aleatorios entre 1 y 9
 2. Mostrar la matriz por pantalla
 3. Mostrar el contenido de una ubicación (fila,columna) ingresada por el usuario
 4. Mostrar los valores de la diagonal principal
 5. Mostrar los valores de la diagonal secundaria
 6. Mostrar el contorno de la matriz
 7. Desea Continuar S/N?

Por ejemplo, para la siguiente matriz de orden 5, se deben mostrar las siguientes salidas según la opción seleccionada por el usuario:

2. Mostrar Matriz

1 6 9 1 4
1 6 4 3 7
5 4 9 1 6
5 1 4 4 1
3 2 9 9 8

3. Ubicación (1, 3) = 3

4. Mostrar Diagonal Principal

1 6 9 4 8

5. Mostrar Diagonal Secundaria

4 3 9 1 3

6. Mostrar Contorno Matriz

1 6 9 1 4
1 7
5 6
5 1
3 2 9 9 8

9. Dada una matriz de orden $n * m$ (donde n y m son valores ingresados por el usuario) realizar un programa que calcule y muestre la suma de los números pares y la suma de los números impares. Mostrar la matriz y los resultados por pantalla.
10. Realizar un programa que calcule la multiplicación de dos matrices de enteros de 3×3 . Inicialice las matrices para evitar el ingreso de datos por teclado.
NOTA: Recuerde para calcular el producto: $c_{ij} = a_{i1} * b_{1j} + a_{i2} * b_{2j} + a_{i3} * b_{3j}$
11. Un cuadrado mágico 3×3 es una matriz 3×3 formada por números del 1 al 9 donde la suma de sus filas, sus columnas y sus diagonales son idénticas. Crear un programa que permita introducir un cuadrado por teclado y determine si este cuadrado es mágico o no. En caso de tratarse de un cuadrado mágico se debe mostrar la suma, caso contrario se debe indicar un mensaje. El programa deberá comprobar que los números introducidos son correctos, es decir, están entre el 1 y el 9. Por ejemplo, en la siguiente matriz las sumas son siempre 15.

2 7 6

9 5 1

4 3 8

12. Se dice que una matriz tiene un punto de silla si alguna posición de la matriz es el menor valor de su fila, y a la vez el mayor de su columna. Escribir un programa que tenga como entrada una matriz de números reales, y calcular la posición de un punto de silla (si es que existe).

13. El proceso de construcción de sillas de oficina se realiza en 5 fases (A,B,C,D,E) que se pueden realizar en 6 máquinas diferentes (1..6), se necesita calcular el costo máximo y mínimo tanto para fases como para máquinas; en la tabla siguiente se muestra la relación costo-proceso-máquina que estos eventos conllevan:

	A	B	C	D	E
1	6	5	9	19	11
2	45	23	21	17	30
3	15	7	23	25	2
4	75	26	38	49	19
5	2	15	76	1	45
6	92	82	4	20	13

14. Construya un programa que lea 3 palabras de mínimo 3 caracteres y máximo 5 caracteres y, a medida que el usuario las va ingresando, construya una “sopa de letras para niños” de tamaño de 20 x 20 caracteres. Para la ubicación de las palabras se debe tener en cuenta las siguientes restricciones:
- Las palabras se ubican al azar en la sopa de letra.
 - Cada una de las palabras debe quedar en una orientación distinta, es decir, una en horizontal, una en vertical y otra en diagonal.
 - Se debe determinar en forma aleatoria el sentido (de izq a derecha o viceversa, de arriba hacia abajo o viceversa) tomando en cuenta la orientación.
 - Las palabras no deben chocar entre sí.

Una vez concluida la ubicación de las palabras, rellene los espacios no utilizados con un número aleatorio del 0 al 9. Finalmente imprima por pantalla la sopa de letras creada.