

Ejercicios UNIDAD VII

Ejercicios con vectores y matrices

1. Ingresar un valor N (< 25). Generar un arreglo de N componentes en el cual las mismas contengan los primeros números naturales pares e imprimirlo.
2. Ingresar un valor entero N (< 30) y a continuación un conjunto de N elementos. Si el último elemento del conjunto tiene un valor menor que 10 imprimir los negativos y en caso contrario los demás.
3. Ingresar un valor entero N (< 20). A continuación ingresar un conjunto VEC de N componentes. A partir de este conjunto generar otro FACT en el que cada elemento sea el factorial del elemento homólogo de VEC. Finalmente imprimir ambos vectores a razón de un valor de cada uno por renglón
4. Ingresar un valor entero N (< 25). A continuación ingresar un conjunto VEC de N componentes. Si la suma de las componentes resulta mayor que cero imprimir las de índice impar, sino los otros elementos.
5. Ingresar un valor entero N (< 30). A continuación ingresar un conjunto UNO y luego otro conjunto DOS, ambos de N componentes. Generar e imprimir otro conjunto TRES intercalando los valores de posición impar de DOS y los valores de posición par de UNO.
6. Ingresar un valor entero N (< 40). A continuación ingresar un conjunto VALOR de N elementos. Determinar e imprimir el valor máximo y la posición del mismo dentro del conjunto. Si el máximo no es único, imprimir todas las posiciones en que se encuentra.
7. Ingresar un valor entero N (< 15). A continuación ingresar un conjunto DATO de N elementos. Generar otro conjunto de dos componentes MEJORDATO donde el primer elemento sea el mayor valor de DATO y el segundo el siguiente mayor (puede ser el mismo si está repetido). Imprimir el conjunto MEJORDATO con identificación.
8. Ingresar un valor entero N (< 25). A continuación ingresar un conjunto GG de N elementos. Imprimir el arreglo en orden inverso generando tres estrategias para imprimir los elementos a razón de: a) Uno por línea, b) Diez por línea, c) Cinco por línea con identificación
9. Ingresar un valor entero N (< 40). A continuación ingresar un conjunto A y luego otro conjunto B ambos de N elementos. Generar un arreglo C donde cada elemento se forme de la siguiente forma: $C[1] = A[1] + B[N]$ $C[2] = A[2] + B[N-1]$
10. Ingresar dos valores enteros M (< 10) y N (< 15). A continuación ingresar un conjunto A de M elementos y luego otro B de N elementos. Generar e imprimir:
a) Un conjunto C resultante de la anexión de A y B.
b) Un conjunto D resultante de la anexión de los elementos distintos de cero de A y B.
11. Ingresar dos valores enteros M (< 25) y N (< 10) A continuación ingresar un conjunto A de M elementos y luego otro B de N elementos, ambos ordenados en forma creciente por magnitud. Generar e imprimir el conjunto TOTAL resultante del apareo por magnitud de los conjuntos A y B.
12. Ingresar un valor entero N (< 40). Luego ingresar un conjunto REFER de N elementos reales

(ingresan ordenados por magnitud creciente). Finalmente ingresar un valor pesquisa X. Desarrollar el programa que determine e imprima:

- a) Con cual elemento (posición) del conjunto coincide, o
- b) Entre cuales dos elementos (posiciones) se encuentra, o
- c) Si es menor que el primero o mayor que el último.

13. Ingresar un valor entero CANT (< 50) y a continuación un conjunto SINOR de CANT elementos. Desarrollar un programa que determine e imprima:

- a) El conjunto SINOR en el que cada elemento original se intercambie por su simétrico: $A[1]$ con $A[CANT]$, $A[2]$ con $A[N-1]$, etc.
- b) El conjunto SINOR ordenado de menor a mayor sobre si mismo indicando la posición que ocupaba cada elemento en el conjunto original.

17. Ingresar dos valores, M (< 30) y N (< 25) y a continuación por filas todos los componentes de una matriz MATRIZA de M filas y N columnas. Desarrollar un programa que:

- a) Imprima la matriz MATRIZA por columnas.
- b) Calcule e imprima el valor promedio de los componentes de la matriz.
- c) Genere e imprima un vector VECSUMCOL donde cada componente sea la suma de la columna homóloga.
- d) Genere e imprima un vector VECMAXFIL donde cada componente sea el valor máximo de cada fila.

18. Ingresar un valor N (< 25) y luego por filas una matriz cuadrada CUADRADA de N filas y columnas. Desarrollar un programa que determine e imprima:

- a) Todos los elementos de la diagonal principal o secundaria según de cual resulte mayor la sumatoria de elementos.
- b) Los elementos del cuarto ($N/2$ filas y $N/2$ columnas) cuya sumatoria resulte mayor (considerando que N fuera par).
- c) Los elementos de la triangular superior o inferior dependiendo de cual tenga mayor sumatoria de elementos.

19. Ingresar dos valores, M (< 20) y N (< 25) y a continuación por columnas todos los componentes de una matriz DESORDE de M filas y N columnas. Desarrollar un programa que:

- a) Ordene (creciente) cada columna de la matriz sobre si misma y la imprima a razón de una columna por renglón.
- b) Ordene (creciente) la matriz sobre si misma por fila desde el elemento 1,1 al M,N y la imprima a razón de una fila por renglón.

20. Ingresar por plano, fila y columna todos los elementos de una matriz MATRIDIM de M planos, filas y columnas. Desarrollar un programa que:

- a) Imprima la matriz MATRIDIM por columnas, fila, plano.
- b) Calcule e imprima el valor promedio de la matriz.
- c) Determine e imprima el mayor valor y en que lugar de la matriz se encuentra.
- d) Genere e imprima una matriz MATCSUMCOL donde cada elemento sea la suma de la columna homóloga.
- e) Genere e imprima una matriz MATMAXFIL donde cada elemento sea el valor máximo de cada fila.

21. Ingresar las edades de los alumnos de un curso e informar cuántos alumnos tienen una edad que supera la edad promedio. Se desconoce la cantidad de alumnos a procesar pero se sabe que a lo sumo se inscriben 50 alumnos en un curso.

22. Dado un arreglo de enteros ($\text{Max} = 1000$) se pide escribir el algoritmo de una rutina que permita mostrar sus elementos pero en orden inverso.

23. Dado un arreglo de enteros ($\text{Max} = 1000$) y otro entero (DATO) se pide desarrollar:

- a) Una rutina que indique si DATO es o no un elemento del arreglo.
- b) Una rutina que indique si DATO es o no un elemento del arreglo y si lo es devuelva la posición de la primera aparición del mismo (Búsqueda Secuencial).
- c) Una rutina que devuelva todas las posiciones en las cuales aparece DATO en el arreglo.
- d) Una rutina que devuelva la posición de la primera y de la última aparición de DATO en el arreglo.

24. Dado un arreglo de enteros ($\text{Max} = 1000$) se pide desarrollar:

- a) Una rutina que devuelva la posición del elemento mayor en el arreglo. Si el mayor aparece varias veces devolver la posición de la primera aparición del mismo.
- b) Una rutina que devuelva todas las posiciones en que aparece el elemento que es el mayor del arreglo (no olvide considerar el caso en que todos los elementos sean iguales)

25. Dado un arreglo de enteros ($\text{Max} = 1000$) ordenado en forma ascendente y otro entero (DATO) se pide desarrollar:

- a) Una rutina que indique si DATO es o no elemento del arreglo y si lo es devuelva la posición de la primera aparición del mismo (Búsqueda Secuencial en Arreglo Ordenado)
- b) Una rutina que indique si DATO es o no elemento del arreglo y si lo es devuelva su posición (Búsqueda Binaria o Dicotómica)

26. Dado un arreglo de enteros ($\text{Max} = 1000$) ordenado en forma ascendente y otro entero (DATO) se pide desarrollar una rutina que inserte DATO en el arreglo de forma ordenada.

27. Dado un arreglo de enteros ($\text{Max} = 1000$) y otro entero (POS) que representa la posición de un elemento del arreglo, se pide desarrollar una rutina que elimine del arreglo el elemento que se encuentra en la posición POS.

28. Dado un arreglo de enteros (Max = 1000) y otro entero (DATO) se pide desarrollar una rutina que elimine del arreglo todos los elementos iguales a DATO, compactándolo.

29. Dado un arreglo de enteros (Max = 1000) se pide desarrollar una rutina que elimine del arreglo aquellos que son múltiplos del elemento que sea menor en el arreglo, compactándolo.

30. Dado un arreglo de caracteres (Max = 1000) se pide desarrollar una rutina que elimine del arreglo todos los elementos repetidos, manteniendo la primera aparición de cada uno de ellos.

31. Ingresar números enteros hasta detectar el fin de datos. Luego, mostrar los 20 mayores ingresados (Ranking)

32. Ingresar los elementos de un arreglo o vector P, a continuación ingresar los de otro arreglo o vector Q. Validar que la cantidad de elementos ingresados para cada uno de los vectores sea la misma y calcular:

- a) La suma y la diferencia de los vectores.
- b) El producto escalar. (Sumatoria de 1 a N de $P_i * Q_i$)

Si la cantidad de componentes no es la misma, no efectuar el cálculo. El máximo de elementos es 30 y son reales.

33. Desarrolle un procedimiento que dados dos arreglos de entrada de N elementos genere un nuevo arreglo de salida donde figuren sólo los elementos comunes a ambos. Ambos arreglos están ordenados en orden ascendente.

34. Ingresar los nombres y los promedios académicos de los alumnos de la carrera de Sistemas. El máximo es de 250 alumnos

- a) Mostrar el listado por pantalla en el orden en que se ingresaron y luego en forma ordenada de menor a mayor.
- b) Mostrar el listado de alumnos cuyo promedio es mayor o igual a 7.

35. Desarrollar el algoritmo de una rutina determine si un arreglo de caracteres donde está cargada una palabra, es o no un palíndromo. Ejemplo:

['N', 'E', 'U', 'Q', 'U', 'E', 'N']

Verdadero ☺ es un palíndromo

['C', 'H', 'U', 'B', 'U', 'T']

Falso ☹ no es palíndromo

extra 1:

- a) Un profesor de programación, recibe un vector, con los minutos de llegada de cada estudiante, respecto de su horario de ingreso. Por ejemplo, si recibe el valor -5 significa que llegó 5 minutos antes. El valor 3 significa que llegó 3 minutos tarde. El valor 0 significa que llegó puntual. Por otro lado, también recibe la cantidad de alumnos que se presentaron, y la cantidad de alumnos que necesitan para abrir el curso. Realizar una función `seAbreCurso` que reciba el array de tiempos, y el mínimo de alumnos, y que devuelva verdadero si se abre el curso o falso si no se abre. Se abre cuando los que llegaron antes o puntuales son iguales o más que los necesarios para abrir.

```
bool seAbreCurso(tiempoLlegada[], cantidadTotal, alumnosMinimosAbrir)
```

ejemplo de valores:

```
tiempoLlegada[] = {-3, 2, 0, 0, 4}
cantidadTotal = 5
alumnosMinimosAbrir = 2
```

Debe devolver: true.

- b) Crear otra función que reciba un array con los minutos de llegada de los 5 días de la semana, la cantidad total que se presentaron cada día, y la cantidad de alumnos necesaria para abrir la clase, y que devuelva un array con verdaderos y falsos según se debió abrir o no cada día. {true, false, false, true, true}

```
seAbreCursoSemana(tiempoLlegadaSemana[] , alumnosTotal, alumnosMinimosAbrir)
```

```
tiempoLlegadaSemana[] = {{-3, 2, 0, 0}, {2, -1, 4, 5}, {2, -1, 4, 5}, {2, -1, 4, 5}, {2, -1, 4, 5} }
alumnosTotal = 4
alumnosMinimosAbrir = 3
```

`tiempoLlegadaSemana` devuelve { true, true, false, true, false }