PRACTICA DIRIGIDA 7 (adicionales)

1. Ingrese la cantidad ***n*** de notas a generar. Genere aleatoriamente un arreglo dinámico de ***n*** notas, es decir, enteros del intervalo [0, 20]. Ahora, ordene dicho arreglo y programe que la computadora seleccione una de esas notas de dicho arreglo e intente adivinar dicha nota, en cada intento fallido la computadora le debe decir si la nota es mayor o menor al valor fallido.

   Ejemplo

   ```
   n     :     4
   Adivine la nota seleccionada    :     12
   La nota seleccionada es mayor.
   Adivine la nota seleccionada    :     16
   La nota seleccionada es mayor.
   Adivine la nota seleccionada    :     18
   La nota seleccionada es menor.
   Adivine la nota seleccionada    :     17
   Adivinaste!
   Notas ordenadas       :       1 7 17 20
   ```

2. Los **puntos silla de montar** (o valor MAXIMIN) en una matriz, son aquellos valores que cumplen la siguiente propiedad:

   Son los valores mínimos de la fila en la que están y a la vez son los valores máximos de la columna donde se encuentran. Así, podemos ver que en la matriz de ejemplo siguiente:

-200	-1300	2000	-1300
-1000	-700	-1500	-1500
800	900	850	800

   El valor MAXIMIN es 800 (Es el valor mínimo de su fila, que a su vez es el máximo de su columna). Además, una matriz se dice que tiene una zona plana si hay por lo menos 2 puntos de silla de montar dentro de ella, como en este caso: 800.

   a. Realice un programa que utilizando funciones permita crear una matriz de F filas por C columnas de forma dinámica. F y C serán ingresados por teclado.
   b. Elabore una función que reciba como parámetros el apuntador doble, F y C y se encargue de ingresar los valores de la matriz.

   c. Averiguar si dicha matriz es plana o no. Si solo tiene un punto de silla de montar, el programa mostrará el mensaje: *Solo tiene un punto de silla de montar*.

      Ejemplo

      ```
      Dimensión de la matriz: 2 3
      Ingrese los valores de la matriz
      Ingrese 3 datos para la fila 1: 2
      3
      4
      ```

Ingrese 3 datos para la fila 2: 5
6
6
Matriz ingresada
2 3 4
5 6 6
El valor minmax es 5
aparece: 1 veces
Matriz con un punto silla de montar

3. A partir de la siguiente matriz:

NOMBRE,	PUESTO,	SALARIO,
VIDAL,	ASISTENTE,	800,
MORALES,	VENTAS,	1600,
SANZ,	VENTAS,	1250,
IGLESIAS,	GERENTE,	2975,
MARTIN,	VENTAS,	1250,
VAZQUEZ,	GERENTE,	2850,
MORENO,	GERENTE,	2450,
JIMENEZ,	ANALISTA,	3000,
GARCIA,	PRESIDENTE,	5000,

Desarrollar un programa en C++, realizar una consulta donde se ingresa el PUESTO como cadena y devolver un reporte en una nueva matriz de NOMBRE Y SALARIO el promedio de salario.

Ejemplo:

VIDAL ASISTENTE 800
MORALES VENTAS 1600
SANZ VENTAS 1250
IGLESIAS GERENTE 2975
MARTIN VENTAS 1250
VAZQUEZ GERENTE 2850
MORENO GERENTE 2450
JIMENEZ ANALISTA 3000
GARCIA PRESIDENTE 5000

- - - - - - -
EL PUESTO ES: VENTAS
- - - - - - -

MORALES 1600
SANZ 1250
MARTIN 1250

El promedio de SUELDO es: 1366.67
---------- - - - - -

Sugerencia:

Donde se puede utilizar la funcion "find":

```
string str ("There are two needles in this haystack."); string str2
("needle");

if (str.find(str2) != string::npos) {
//.. found.
}
```

4. Usando los conceptos de asignación dinámica de memoria, desarrolle código en C++ para una multiplicación entre una matriz y un vector.

Ejemplo

Ingrese el número de filas en la matriz 2
Ingrese el número de columnas en la matriz 3
Ingrese los elementos de la matriz A

2
6
4
1
4
8

Ingrese los elementos del vector x 3
5
2

*A*x =*

44
39

5. Ingrese la cantidad de filas y columnas de una matriz de notas a generar (aleatoriamente). Luego, imprima dicha matriz y ordene cada una de sus columnas de mayor a menor. Finalmente, imprima la matriz ordenada.

Ejemplo

filas 2
columnas 3

2 3 0
20 3 6

Matriz de notas con las columnas ordenadas:

20 3 6
2 3 0

6. Observe la matriz:

15	0	0	22	0	-15
0	11	33	0	0	0
0	0	0	-6	0	0
0	0	0	0	0	0
91	0	0	0	0	0
0	0	28	0	0	0

Esta matriz se dice **rala** porque solo ocho de sus 36 elementos son distintos de cero, es decir, podríamos indicar que una matriz rala es aquella que tiene muchos ceros (más de un 75% del total). Para reducir el espacio de memoria que ocupa esta matriz se crea una nueva matriz que conserva la posición y el valor de los elementos no nulos. Esta matriz tiene la siguiente estructura:

Consta de solo tres columnas.

Tiene ***n+1*** filas, donde n es el número de los elementos no nulos de la matriz original.
La primera fila de la matriz resultante está formada por el número de filas, el número de columnas y la cantidad de elementos no nulos de la matriz original.

Las restantes filas contienen:

En las dos primeras columnas, la posición donde se encuentra el elemento no nulo. En la tercera columna, el elemento no nulo de la primera matriz.

La matriz reducida correspondiente a la matriz inicial es:

6	6	8
0	0	15
0	3	22
0	5	-15
1	1	11
1	2	33
2	3	-6
4	0	91
5	2	28

Realice un programa que:

a) Utilizando funciones permita crear una matriz de F filas por C columnas de forma dinámica. F y C serán ingresados por teclado.
b) Determine si es conveniente o no reducirla.
c) Si se justifica, reserve memoria para generar la matriz reducida.
d) Imprima la matriz reducida.

7. Dadas las declaraciones: int

   stack_arr[5];
   int* heap_arr;

   Escribir código en C++ para:

   1. Asignar dinámicamente 5 enteros del heap(montículo) para heap_arr
   2. Para cada arreglo (stack_arr, heap_arr):

   a. Setear el ítem en el índice 2 a 42
   b. Imprimir el item en el índice 2
   c. Incrementar el item en el índice 2
   d. Imprimir la dirección del primer item

   3. Liberar la memoria asociada con heap_arr

   La salida debe ser:

   Arreglo Stack: 0, 0, 0, 0, 0
   Arreglo Heap: El puntero al arreglo es NULL. Arreglo
   Heap: 0, 0, 0, 0, 0
   Arreglo Stack: 0, 0, 42, 0, 0
   Arreglo Heap: 0, 0, 42, 0, 0
   Stack: 42
   Heap: 42
   Arreglo Stack: 0, 0, 43, 0, 0
   Arreglo Heap: 0, 0, 43, 0, 0
   El arreglo en el Stack se inicia en: 0x7ffc20b37460 El arreglo en el Stack se inicia en: 0x7ffc20b37460 El arreglo en el Heap se inicia en: 0x205fc50
   El arreglo en el Heap se inicia en: 0x205fc50

8. Utilizando ***new***, se tiene un vector de tamaño inicial 1 y se ingresan valores cada vez que se llena el vector se duplica su tamaño para ingresar más valores y luego en un vector de tamaño 4 quedarse con los 4 mayores elementos del vector mayor.

   Por ejemplo ingresando los elementos

   5,8,9,6,3,8, ingresar valor: 5
   dimension tabla 1
   [0]5,
   Vector de Tamaño Cuatro con los Mayores [0]5,
   ---

   ingresar valor: 8
   dimension tabla 2
   [0]5, [1]8,
   Vector de Tamaño Cuatro con los Mayores [0]8,
   [1]5,

---

ingresar valor: 9
dimension tabla 4
[0]5, [1]8, [2]9,
Vector de Tamaño Cuatro con los Mayores [0]9,
[1]8, [2]5,

ingresar valor: 6
dimension tabla 4
[0]5, [1]8, [2]9, [3]6,
Vector de Tamaño Cuatro con los Mayores [0]9,
[1]8, [2]6, [3]5,

---

ingresar valor: 3
dimension tabla 8
[0]5, [1]8, [2]9, [3]6, [4]3,
Vector de Tamaño Cuatro con los Mayores [0]9,
[1]8, [2]6, [3]5,

---

ingresar valor: 8
dimension tabla 8
[0]5, [1]8, [2]9, [3]6, [4]3, [5]8,
Vector de Tamaño Cuatro con los Mayores [0]9,
[1]8, [2]8, [3]6,
:

Sugerencia: Si cree necesario puede usar la función memcpy() cuya sintaxis es:

void memcpy(void* dest, const void* src,size_t count);*

La función memcpy () toma tres argumentos: *dest*, *src* y *count*. Esta función, cuando se llama, copia el recuento de bytes de la ubicación de memoria apuntada por *src* a la posición de memoria apuntada por *dest*.