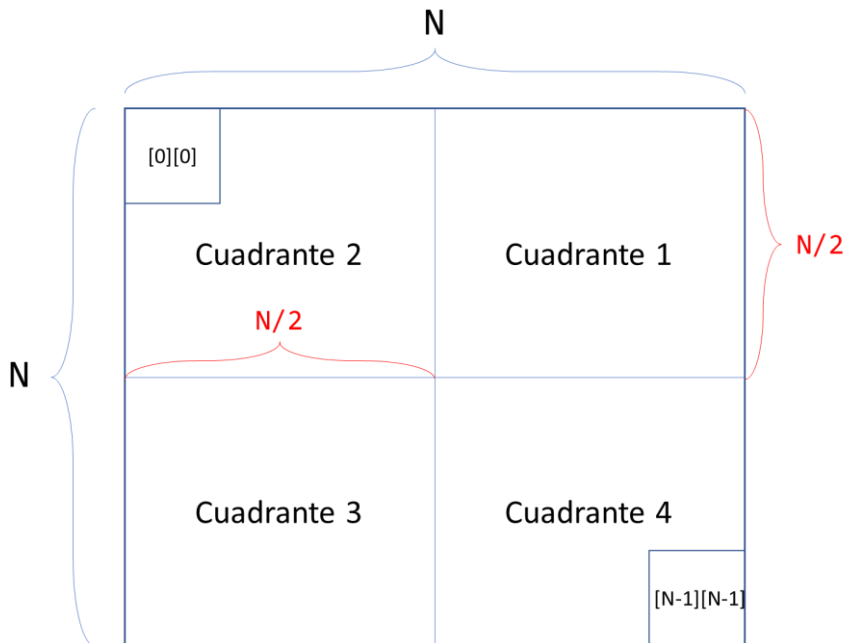


Universidad Nacional de Asunción - Facultad de Ingeniería
Cátedra de Computación – 2do Examen Final – 09/07/2021 – Tema 2

En este ejercicio trabajaremos con una forma particular de ordenamiento de una matriz cuadrada de $N \times N$ elementos enteros. Se debe implementar un programa en C++ que solicite por teclado el valor de N (se debe validar que N sea entero, par y mayor que 3), y que cargue los elementos de la matriz cuadrada con valores aleatorios entre 0 y $2*N-1$. Para el ordenamiento de los los elementos de la matriz, ésta se dividirá en cuatro submatrices cuadradas de igual tamaño, a las que llamaremos “cuadrantes” (como se muestra en la figura).



El programa deberá ordenar los elementos de cada cuadrante en forma ascendente (de la forma que se muestra en el ejemplo). Además, deberá mostrar en pantalla la matriz original y la matriz resultante.

Por último, para cada cuadrante se deberá calcular la suma de sus elementos, y mostrar en pantalla el mayor valor entre tales sumas.

Ejemplo (con $N=6$):

Matriz inicial

5	4	7	9	3	10
1	0	7	5	5	2
2	6	3	6	3	4
5	10	1	7	4	3
1	2	0	0	2	1
4	3	5	8	4	4



Matriz final

0	1	2	2	3	3
3	4	5	4	5	5
6	7	7	6	9	10
0	1	1	0	1	2
2	3	4	3	4	4
5	5	10	4	7	8

La mayor de las sumatorias es: 47

Distribución de puntajes:

- Carga por teclado y validación de N : 1p
- Carga de elementos de la matriz (valores aleatorios): 2p
- Ordenamiento correcto de cada cuadrante e impresión de la matriz (inicial y final): 9p
- Cálculo de la suma de cada cuadrante: 1p
- Obtención e impresión en pantalla del mayor valor entre las sumas: 2p

Universidad Nacional de Asunción - Facultad de Ingeniería

Computación – 2021c1

Segundo Examen Final - Tema 3

Fecha: 09/07/2021

TEMA 3: Puede ser realizado en DevC++ o similar • El código deberá ser copiado o alzado al VPL (uno de los 2, NO ambos). Si elige alzar, solo puede hacerlo una vez, se recomienda alzar cuando ya no modificará el tema. No se contarán con casos de prueba • Será corregido manualmente por los docentes de la cátedra.

El nombre del archivo debe ser T3F2_numeroCI seguido de la extensión .cpp. **Ejemplo** T3F2_1234567.cpp

Debe realizar uno de los 2 temas prácticos, solo uno.

Juego de Bingo. 15p

En el juego del bingo los números son escogidos al azar de un conjunto de 75 bolas llamado bolillero, generalmente de 1 a 75. Los números en el rango de 1 a 15 están asociados con la letra B, del 16 al 30 con la letra I, del 31 al 45 con la letra N, del 46 al 60 con la letra G, y del 61 a 75 con la letra O.

La persona que dirige el juego selecciona una bolilla al azar, del bolillero, y luego anuncia la letra y el número. A continuación, guarda la bolilla extraída para que no se pueda utilizar de nuevo en ese juego.

Se debe crear un **struct Bolilla** con los siguientes campos: char letra, unsigned short numero, que deberá utilizarse para resolver el problema propuesto (**1p**).

Se pide crear una aplicación que simule el juego del Bingo. La app debe mostrar el siguiente menú y que se repita mientras NO se presione la opción 4 (Salir) (**1p**).

B	I	N	G	O
9	25	34	48	69
15	19	31	59	74
2	28	FREE	52	62
7	16	41	58	70
4	20	38	47	64

En el lugar correspondiente a FREE debe ir la bola de bingo O, letra F

1. Crear bolillero	Crear un vector para almacenar las bolillas. Imprimir el bolillero, las bolillas de cada letra en una fila (2p)
2. Generar cartón	Crear una matriz para almacenar las bolillas. Las bolillas deben generarse aleatoriamente (1p), y no deben repetirse (1p) (tenga en cuenta que debe generar números correspondientes para cada columna) (1p). Colocar la bolilla 0 en la posición mostrada en la imagen (FREE). Imprimir el cartón generado (1p)
3. Girar bolillero	Mezclar las bolillas en el bolillero, deberá utilizar la función rand(). Imprimir el bolillero (2p)
4. Extraer bolilla	Debe sacar una bolilla del bolillero, mostrar en pantalla e indicar si está o no en el cartón (2p).
5. Salir	Cierra la aplicación.

Cada opción debe ser resuelto en una función (**2p**). En el **main** solo deben crearse las variables e invocar a las funciones (**1p**).

Ejemplos:

Al seleccionar la opción 1:

```
B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15
I16 I17 I18 I19 I20 I21 I22 I23 I24 I25 I26 I27 I28 I29 I30
N31 N32 N33 N34 N35 N36 N37 N38 N39 N40 N41 N42 N43 N44 N45
G46 G47 G48 G49 G50 G51 G52 G53 G54 G55 G56 G57 G58 G59 G60
O61 O62 O63 O64 O65 O66 O67 O68 O69 O70 O71 O72 O73 O74 O75
```

Al seleccionar la opción 2: <div>B8 I20 N34 G56 065 B6 I29 N33 G46 072 B10 I24 F0 G51 074 B2 I25 N40 G55 067 B14 I23 N41 G48 062</div>	Al seleccionar la opción 4: <div>La bolilla I26 NO esta en el carton</div>
---	---

Al seleccionar la opción 3:

I26 I22 063 I23 N36 G60 062 G58 G52 067 G46 N45 I20 073 I17
B10 069 070 N41 072 N43 G48 N35 B9 G55 I28 066 N39 G49 064
B6 G56 I18 B7 075 I29 I16 N44 N37 G51 I30 G47 G54 N42 B2
G50 065 B3 074 N32 071 I24 N34 I25 061 I21 B1 N40 B5 B13
I27 G59 B11 N33 068 B8 G57 B14 I19 N31 B4 B15 B12 N38 G53