

Indicaciones específicas:

- Esta evaluación contiene 11 páginas (incluyendo esta página) con 3 preguntas. El total de puntos son 20.
- Crea una carpeta llamada: PC2
- Luego crea una subcarpeta para cada pregunta:
 1. p1
 2. p2
 3. p3
- Una vez que termines de realizar las 3 preguntas:
 1. Elimina el directorio cMake-Buil-debug de cada pregunta
 2. Comprime la carpeta PC2 usando el winzip y obtendrás el archivo PC2.zip
- Sube el archivo **PC2.ZIP** al gradescoupe a www.gradescope.com.
- **Recuerda que solo se calificará si has enviado en el formato indicado.**

Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación
 - Aplicar conocimientos de computación apropiados para la solución de problemas definidos y sus requerimientos en la disciplina del programa. (nivel 2)
 - Diseñar, implementar y evaluar soluciones a problemas complejos de computación.(nivel 2)
 - Crear, seleccionar, adaptar y aplicar técnicas, recursos y herramientas modernas para la práctica de la computación y comprende sus limitaciones. (nivel 2)
- Para los alumnos de las carreras de Ingeniería
 - Aplicar conocimientos de ingeniería en la solución de problemas complejos de ingeniería (nivel 2).
 - Diseñar soluciones relacionados a problemas complejos de ingeniería (nivel 2)

Crear, seleccionar y utilizar técnicas, habilidades, recursos y herramientas modernas de la ingeniería y las tecnologías de la información, incluyendo la predicción y el modelamiento, con la comprensión de sus limitaciones (nivel 2)

- Para los alumnos de Administración y Negocios Digitales

Analizar información verbal y/o lógica proveniente de distintas fuentes, encontrando relaciones y presentándola de manera clara y concisa (nivel 2)

Analizar y evaluar el comportamiento del consumidor y el desarrollo de estrategias comerciales (nivel 2)

Trabajar de manera efectiva con equipos multidisciplinarios y diversos en género, nacionalidad, edad, etc. (nivel 2)

Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	7	
2	6	
3	7	
Total:	20	

1. (7 points) **Evalua matrices dinámicas**

Una imagen en tonos de grises, se puede representar en una matriz utilizando números desde 0 al 9. Donde 0 representa gris claro y el 9 gris oscuro. Se pide que diseñe e implemente un programa que permita:

- Leer el orden de una matriz y genere la matriz con valores aleatorios cuyo rango de valores sea desde el 0 al 9
- Imprima la matriz
- Cree una segunda matriz, que representa el revelado de la imagen, para lo cual cada número de la matriz de números se represente por un caracter. El caracter se signa considerando lo siguiente:

Número	Caracter
0	.
1,2,3	-
4,5,6	#
7,8,9	*

- Imprima la segunda matriz.
- **Para resolver la pregunta es indispensable que utilice matrices dinámicas y que haga un buen uso de la zona de memoria Heap**

Para asignar el puntaje a esta pregunta es necesario que las dos matrices nuevas se creen y no solo sean impresas.

Algunos ejemplos de diálogo de este programa serían:

Listing 1: Ejemplo 1

```

Filas: 6
Columnas : 8

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

-  #  *  #  -  #  #  -
*  -  -  *  .  *  -  #
.  #  -  #  -  *  *  *
-  .  -  -  *  #  *  -
-  *  *  *  -  #  -  -
*  -  -  *  #  *  *  #

```

Listing 2: Ejemplo 2

```

Filas: 10
Columnas : 12

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

-  #  *  #  -  #  #  -  *  -  -  *
.  *  -  #  .  #  -  #  -  *  *  *
-  .  -  -  *  #  *  -  -  *  *  *
-  #  -  -  *  -  -  *  #  *  *  #
#  .  -  #  -  .  #  -  -  .  #  -
#  #  #  *  #  #  #  *  -  *  #  #
-  #  #  *  #  #  -  .  *  *  #  *
*  #  -  -  #  *  -  .  #  *  *  -
#  #  #  *  #  .  #  *  *  -  *  -
*  -  -  #  -  .  #  -  #  *  #  *

```

Listing 3: Ejemplo 3

```

Filas: 5
Columnas : 5

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

-  #  *  #  -
#  #  -  *  -
-  *  .  *  -
#  .  #  -  #
-  *  *  *  -

```

Listing 4: Ejemplo 1

Filas: 15

Columnas : 15

```

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

```

```

-  #  *  #  -  #  #  -  *  -  -  *  .  *  -
#  .  #  -  #  -  *  *  *  -  .  -  -  *  #
*  -  -  *  *  *  -  #  -  -  *  -  -  *  #
*  *  #  #  .  -  #  -  .  #  -  -  .  #  -
#  #  #  *  #  #  #  *  -  *  #  #  -  #  #
*  #  #  -  .  *  *  #  *  *  #  -  -  #  *
-  .  #  *  *  -  #  #  #  *  #  .  #  *  *
-  *  -  *  -  -  #  -  .  #  -  #  *  #  *
.  *  -  *  *  -  -  #  *  *  *  #  *  -  #
#  #  -  *  #  *  -  -  *  -  *  .  *  *  #
.  *  #  -  *  #  #  -  -  #  *  *  #  *  -
.  -  .  #  #  #  #  *  #  -  -  *  #  *  #
-  -  *  *  #  *  #  -  *  #  *  *  #  -  *
*  -  -  *  *  #  #  -  -  -  *  #  .  #  *
*  *  *  *  *  #  #  -  .  -  .  *  -  #  #

```

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (3pts)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (2pts)	El diseño tiene algunas deficiencias pero la ejecución es correcta (1pts).	El diseño es deficiente y la ejecución no es correcta (0.5pts)
Sintaxis	No existen errores sintácticos o de compilación (2pts)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (1.5pts).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (1pts).	El código tiene errores de sintaxis que afectan el resultado (0.5pts)
Optimizacion	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2pts)	El código es de buen performance durante la ejecución (1.5pts)	El código no está optimizado pero la ejecución no es deficiente(1pts)	El código no está optimizado y la ejecución es deficiente (0pts)

2. (6 points) **Vectores**

Desarrolle un programa que permita generar un vector con mas de 20 números enteros generados al azar. Los números al azar estarán entre 1 y 9. Luego el programa realice lo siguiente:

- Imprimir el vector generado
- Solicitar al usuario un número entre 1 y 9 que será el dato a buscar y cuyo valor debe validar.
- Formar un nuevo vector, con las posiciones en que el dato a buscar aparece en el primer vector.
- Imprimir el vector de posiciones.
- Para asignar el puntaje a esta pregunta es **necesario que utilice vectores**.

Listing 5: Ejemplo

```
Numero mayor a 20 : 30
Vector con datos aleatorios :
7  5  8  2  3  4  8  6  6  2  2  6  2  8  1  2  1  3  2  1
   3  3  6  6  1  1  7  8  9  4

Numero entre 1 - 9 : 8

Vector de ocurrencias
2  6  13  27
```

Listing 6: Ejemplo

```
Numero mayor a 20 : 15
Numero mayor a 20 : 25
Vector con datos aleatorios :
4  8  2  2  5  1  6  2  2  4  2  6  9  1  8  1  9  9  8  6
   2  2  1  4  5

Numero entre 1 - 9 : 4

Vector de ocurrencias
0  9  23
```

Listing 7: Ejemplo

```
Numero mayor a 20 : 35
Vector con datos aleatorios :
1  2  5  3  8  8  5  7  7  5  2  5  7  3  7  1  8  6  5  1
   1  1  5  3  8  3  2  3  4  1  3  6  1  5  9
```

```
Numero entre 1 - 9 : 1

Vector de ocurrencias
0  15  19  20  21  29  32
```

Listing 8: Ejemplo

```
Numero mayor a 20 : 21
Vector con datos aleatorios :
5  7  4  6  1  3  1  2  2  6  3  4  6  1  6  2  2  4  5  1
  8

Numero entre 1 - 9 : 2

Vector de ocurrencias
7  8  15  16
```

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (3pts)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (2pts)	El diseño tiene algunas deficiencias pero la ejecución es correcta (1pts).	El diseño es deficiente y la ejecución no es correcta (0.5pts)
Sintaxis	No existen errores sintácticos o de compilación (2pts)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (1.5pts).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (1pts).	El código tiene errores de sintaxis que afectan el resultado (0.5pts)
Optimizacion	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2pts)	El código es de buen performance durante la ejecución (1.5pts)	El código no está optimizado pero la ejecución no es deficiente(1pts)	El código no está optimizado y la ejecución es deficiente (0pts)

3. (7 points) **POO**

Desarrolle un programa orientado a objetos, que permita leer como dato un número de al menos 3 dígitos y el programa a través de los servicios que ofrece la clase CNumero, permita imprimir:

- Número de dígitos que tiene el número
- La suma de los dígitos
- El número invertido
- Todos los divisores menores al número.
- **Para asignar el puntaje a esta pregunta es necesario que haya realizado un programa orientado a objetos. Es decir utilice clases y objetos.**

Listing 9: Ejemplo 4

```
Numero de al menos 3 digitos : 876

Analizando el numero
Numero de digitos           : 3
La suma de sus digitos es   : 21
El numero invertido es      : 678
Los divisores menores al numero son:
1   2   3   4   6   12   73   146   219   292   438
```

Listing 10: Ejemplo 4

```
Numero de al menos 3 digitos : 123456

Analizando el numero
Numero de digitos           : 6
La suma de sus digitos es   : 21
El numero invertido es      : 654321
Los divisores menores al numero son:
1   2   3   4   6   8   12   16   24   32   48   64   96
    192   643   1286   1929   2572   3858   5144   7716
    10288   15432   20576   30864   41152   61728
```

Listing 11: Ejemplo 4

```
Numero de al menos 3 digitos : 4321

Analizando el numero
Numero de digitos           : 4
La suma de sus digitos es   : 10
El numero invertido es      : 1234
```

```
Los divisores menores al numero son:  
1    29    149
```

A continuación se muestra parte del código que Ud. debe completar, de tal manera que el programa realice lo que se pide.

Listing 12: Ejemplo

```
#include "CNumero.h"  
  
int main()  
{  
    tNumero numero;  
    do{  
        cout << "Numero de al menos 3 digitos: ";  
        cin >> numero;  
    }while(numero<100);  
  
    CNumero obj(numero);  
  
    cout << "\nAnalizando el numero" << "\n";  
    cout<<"Numero de digitos: " << obj.numDeDigitos() << "\n";  
    cout<<"La suma de sus digitos es: " << obj.sumaDeDigitos()  
        << "\n";  
    cout<<"El numero invertido es: " << obj.numInvertido()<< "\n";  
    cout<<"Los divisores menores al numero son: \n";  
    vector<tNumero> vecDiv = obj.divisores();  
    for(const auto &item:vecDiv)  
        cout << item << "   ";  
    return 0;  
}
```

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (3pts)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (2pts)	El diseño tiene algunas deficiencias pero la ejecución es correcta (1pts).	El diseño es deficiente y la ejecución no es correcta (0.5pts)
Sintaxis	No existen errores sintácticos o de compilación (2pts)	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (1.5pts).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (1pts).	El código tiene errores de sintaxis que afectan el resultado (0.5pts)
Optimizacion	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2pts)	El código es de buen performance durante la ejecución (1.5pts)	El código no está optimizado pero la ejecución no es deficiente(1pts)	El código no está optimizado y la ejecución es deficiente (0pts)