

### Indicaciones específicas:

- Esta evaluación contiene 7 páginas (incluyendo esta página) con 3 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Cada pregunta deberá ser respondida en un solo archivo con el número de la pregunta.
  - p1.cpp
  - p2.pdf
  - p3.cpp
- Deberás subir estos archivos directamente a [www.gradescope.com](http://www.gradescope.com), uno en cada ejercicio. También puedes crear un .zip

## Calificación:

Tabla de puntos (sólo para uso del professor)

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

1. (6 points) **Contenedores y Complejidad Algorítmica**

Dado un vector de números enteros, determine cuántos elementos nunca se repiten en él. La complejidad de su solución debe ser de  $O(n)$ . Utilice el siguiente vector para comprobar su resultado:

```
int main(){
    vector<int> vec = {1,5,3,2,4,1,2,3,4,8,11};
    int resultado = elementos_no_repetidos(vec);
    cout << "El resultado es:" << resultado << endl;
}
```

Salida:

```
El resultado es: 3 // 5,8,11
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Librería Estandar	Selección del contenedor de acuerdo con lo solicitado, uso adecuado de los iteradores, estructuras genéricas basados en contenedores. (3pts)	Selección del contenedor correcto, estructuras genéricas basados en contenedores. (2pts)	Selección del contenedor correcto, estructuras genéricas basados en contenedores, errores en el funcionamiento pasa algunas pruebas. (1pts).	No se selección ni el contenedor ni se desarrolló algoritmos y estructuras genéricas. (0pts)
Complejidad Algorítmica	Buen nivel de abstracción, el problema logro realizar con la complejidad algorítmica solicitado, funciona correctamente y sin errores. (3pts)	Buen nivel de abstracción, el problema logro realizar lo solicitado sin lograr alcanzar la complejidad algorítmica solicitado, funciona correctamente y sin errores. (2pts)	Programa no funciona adecuadamente, bajo nivel de abstracción, más de 3 errores, nivel de complejidad algorítmica incorrecta. (1pts)	Se intento pero no se logró que funcione lo solicitado. (0pts)

## 2. (7 points) Invariante de Bucle

La potenciación con exponente natural  $n$  se puede calcular multiplicando  $n$  veces la base:

$$x^n = \underbrace{x \times x \times \cdots \times x}_{n \text{ veces}}.$$

Sin embargo, existe un algoritmo el cual permite calcular la misma operación en menos pasos ó menos operaciones:

1. `PotenciaRapida( $x, n$ )`
2. `if  $n == 1$`
3. `return  $x$`
4. `else if  $n \% 2 == 1$`
5. `return  $x \cdot \text{PotenciaRapida}(x, n - 1)$`
6. `else`
7. `$p = \text{PotenciaRapida}(x, n/2)$`
8. `return  $p \times p$`

Para este problema:

- Describa el Invariante de Bucle.
- Utilice el Invariante de Bucle para demostrar el algoritmo:
  - Inicialización: ¿El I.B. se cumple en la primera iteración? ¿Como?
  - Mantenimiento: Asumiendo que el I.B. se cumple al comenzar la iteración  $i$ , ¿Este se mantiene al terminar la iteración? ¿Como?
  - Terminación: Al terminar el bucle, utilice el I.B. para demostrar el algoritmo.
- Envíe su solución con el nombre de la pregunta, el formato puede ser PDF o imagen. No es necesario implementar el algoritmo en C++, pero puede hacerlo para verificarlo.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Complejidad Algorítmica	Buen nivel de abstracción, el problema logro realizar con la complejidad algorítmica solicitado, funciona correctamente y sin errores. (7pts)	Buen nivel de abstracción, el problema logro realizar lo solicitado sin lograr alcanzar la complejidad algorítmica solicitado, funciona correctamente y sin errores. (5pts)	Programa no funciona adecuadamente, bajo nivel de abstracción, más de 3 errores, nivel de complejidad algorítmica incorrecta. (3pts)	Se intento pero no se logró que funcione lo solicitado. (1pts)

### 3. (7 points) **Contenedores y Programación Concurrente**

Implementa un programa que simule el funcionamiento de una red social. El programa debe tener las siguientes características:

- Crear la clase **Usuario** que almacene la información de cada usuario: nombre, correo, contraseña, lista de amigos y lista de publicaciones.
- Crear la clase **Publicacion** la cual almacene la información de cada publicación: autor, fecha, contenido y lista de comentarios.
- Crear la clase **Comentario** que almacene la información de cada comentario: autor, fecha y contenido.
- Usar un **contenedor asociativo** para almacenar las publicaciones realizadas, usando el autor y la fecha como clave. La complejidad para esta operación debe ser la mínima posible.
- El programa debe usar la librería **thread** para crear varios hilos que simulen los comentarios de los usuarios en una determinada publicación. Cada hilo debe recibir como parámetro el correo de un usuario, fecha y contenido del comentario.
- El programa debe sincronizar el acceso a los contenedores asociativos usando **mutex**. Por ejemplo, si un hilo quiere crear una publicación, debe bloquear el acceso al contenedor de publicaciones hasta que termine de insertarla.
- Si un hilo quiere comentar una publicación, debe esperar a que haya una publicación disponible y luego bloquear el acceso al contenedor de publicaciones hasta que termine de insertar el comentario.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Librería Es-tandar	Selección del contenedor de acuerdo con lo solicitado, uso adecuado de los iteradores, estructuras genéricas basados en contenedores. (3pts)	Selección del contenedor correcto, estructuras genéricas basados en contenedores. (2pts)	Selección del contenedor correcto, estructuras genéricas basados en contenedores, errores en el funcionamiento pasa algunas pruebas. (1pts).	No se selección ni el contenedor ni se desarrolló algoritmos y estructuras genéricas. (0pts)
Programación Concurrente	Buen nivel de abstracción, el problema se desarrolla utilizando la cantidad de hilos solicitados, se controla adecuadamente los race condition, funciona correctamente y sin errores. (4pts)	Buen nivel de abstracción, el problema no se utiliza la cantidad de hilos solicitados, no se controla los race condition adecuadamente, funciona correctamente y sin errores. (3pts)	Programa no funciona, bajo nivel de abstracción, más de 3 errores visibles , no se usa los hilos adecuadamente ni un control de race condition. (2pts)	Contiene errores que no hace que funcione el programa. (1pts)