

Indicaciones específicas:

- Esta evaluación contiene 5 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, uno en cada ejercicio. También puedes crear un .zip

Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	6	
2	8	
3	6	
Total:	20	

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

Dado un `forward_list<int>` no vacío, de N números enteros aleatorios. Implemente un algoritmo que permita revertir el orden de los elementos, es decir:

- El primer elemento pasará a ser último y el último el primero.
- El segundo elemento será el penúltimo y el penúltimo el segundo, y así con los demás elementos.

Implemente su algoritmo en C++, y utilizando la notación Big O indique su complejidad algorítmica. 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)
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. (8 points) **Invariante de Bucle**

El siguiente algoritmo toma como entrar dos números enteros positivos a y b , y retorna un resultado que depende de ambos. Su objetivo es determinar el resultado y demostrarlo.

FOO(a, b)

```

1.   $r = 1$ 
2.  while( $b > 0$ )
3.      if( $b \% 2 == 1$ )
4.           $res = res \times a$ 
5.           $a = a \times a$ 
6.           $b = b/2$       // Division entera
7.  return   $r$ 
```

Para este algoritmo:

- (a) (2 pts.) Indique que retorna el algoritmo **FOO**, es decir como se relaciona el resultado final con a y b . Sea directo y conciso.
- (b) (2 pts.) Para demostrar el algoritmo usted debió definir un Invariante de Bucle. Describa el Invariante de Bucle, detalle su respuesta.
- (c) (4 pts.) Demuestre la correctitud del algoritmo utilizando el Invariante de Bucle definido en (b).

Envíe su solución con el nombre de la pregunta, el formato puede ser PDF o imagen. No es necesario implementar el algoritmo en C++.

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. (8pts)	Buen nivel de abstracción, el problema logro realizar lo solicitado sin lograr alcanzar la complejidad algorítmica solicitado, funciona correctamente y sin errores. (6pts)	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. (6 points) **Expresiones Lambda y Programación Concurrente**

Un Red Neuronal está compuesta de una entrada $\mathbf{x} \in \mathbb{R}^n$ y una salida $\mathbf{y} \in \mathbb{R}^m$, donde:

$$\mathbf{y} = g(\mathbf{z}) = g(W\mathbf{x} + \mathbf{b}), \quad g(\mathbf{z}) = \frac{1}{1 + e^{-\mathbf{z}}},$$

donde $W \in \mathbb{R}^{m \times n}$ y $\mathbf{b} \in \mathbb{R}^m$. Para implementar esta neuronal considere los siguiente:

- Los elementos de \mathbf{x}, W deben ser números reales aleatorios entre 0 y 1. Por otro lado, \mathbf{b} debe estar conformado por números reales aleatorios entre -0.1 y 0.1 . **Utilice únicamente expresiones Lambda para generar estas variables.**
- Utilice la librería thread para paralelizar el cálculo de \mathbf{y} .
- La función g se aplicar elemento-por-elemento. Es decir como $\mathbf{z} \in \mathbb{R}^m$, $g(\mathbf{z}) \in \mathbb{R}^m$.
- Implemente este mecanismo para $n = 10000$ y $m = 100$. El uso de vectores o matrices dinámicas es irrelevante.

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. (3pts)	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. (2pts)	Programa no funciona, bajo nivel de abstracción, más de 3 errores visibles, no se usa los hilos adecuadamente ni un control de race condition. (1pts)	Contiene errores que no hace que funcione el programa. (0pts)