

Indicaciones específicas:

- Esta evaluación contiene 6 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.cpp
 - p3.cpp
- Deberás subir estos archivos directamente a www.gradescope.com, uno en cada ejercicio. También puedes crear un .zip

Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación
 - Aplicar conocimientos de computación y de matemáticas apropiadas para la disciplina. (Evaluar)
 - Analizar problemas e identificar y definir los requerimientos computacionales apropiados para su solución.(Usar)
 - Utilizar técnicas y herramientas actuales necesarias para la práctica de la computación. (Usar)
- Para los alumnos de las carreras de Ingeniería
 - Capacidad de aplicar conocimientos de matemáticas (nivel 3)
 - Capacidad de aplicar conocimientos de ingeniería(nivel 2)
 - Capacidad para diseñar un sistema, un componente o un proceso para satisfacer las necesidades deseadas dentro de restricciones realistas (nivel 2)

Calificación:

Tabla de puntos (sólo para uso del professor)

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

1. (6 points) **Librería Estandar**

Genere un vector con 1000 números enteros aleatorios entre 1 y 99.

- Utilizar la librería `algorithm` para generar los números aleatorios del 1 al 99.
- Implemente una función que permita imprimir los elementos de un vector en una línea.
- Genere un mapa con los números aleatorios generados como claves. Los valores deben indicar si el número es primo (`true`) o computesto (`false`).
- Genere otro mapa con los números aleatorios generados como claves y las veces que se repiten como valores. Luego, imprima los números de acuerdo a las repeticiones (descendente).

Algunos ejemplos de diálogo de este programa serían:

Listing 1: Ejemplo 1

```
vector generado... Done.

Numero      Primo
    4      false
    5       true
    6      false
    .
    .
    .

Orden:
6    ->   10 veces
4    ->    5 vez
5    ->    3 veces
    .
    .
    .
```

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. (6pts)	Selección del contenedor correcto, estructuras genéricas basados en contenedores. (4pts)	Selección del contenedor correcto, estructuras genéricas basados en contenedores, errores en el funcionamiento pasa algunas pruebas. (2pts).	No se selección ni el contenedor ni se desarrolló algoritmos y estructuras genéricas. (0pts)

2. (7 points) **Complejidad Algorítmica**

Listing 2: Algoritmo 1

```

int foo(int a, int b){
    while (b > 0){
        int q = a / b;
        int r = a - q * b;

        a = b;
        b = r;
    }

    return a;
}

```

La función `foo`, en el Algoritmo 1, retorna un número entero (a y b no-negativos). Dentro de el archivo `p2.cpp`:

- A modo de comentario, indique que representa la salida de la función con respecto a los valores de `a` y `b`. De una explicación de su respuesta.
- Imprima el *Invariante de Bucle* del Algoritmo 1 en cada iteración.

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) **Programación Concurrente**

Utilizando matrices dinámicas, implemente un programa que permita generar dos matrices A y B con números aleatorios entre 0 y 99. El programa debe solicitar la columna y fila para A , **necesariamente la columna y fila deben ser distintos**. Si A es una matriz de $m \times n$, entonces automáticamente B debe ser de $n \times m$. Luego calcule

$$X = A \times B + \text{mean}(A),$$

donde **mean** calcula el promedio de todos los elementos de A , y el operador $+$ adiciona **mean**(A) a cada elemento de $A \times B$. Calcule X de dos formas:

- Paralelizando con respecto a los datos. Cada hilo debe operar a través de una parte de las matrices.
- Paralelizando con respecto a las tareas. Un hilo debe calcular $A \times B$ y otro **mean**(A).

Utilice únicamente **dos hilos** para este problema.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
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. (7pts)	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. (5pts)	Programa no funciona, bajo nivel de abstracción, más de 3 errores visibles, no se usa los hilos adecuadamente ni un control de race condition. (3pts)	Contiene errores que no hace que funcione el programa. (1pts)