

Indicaciones específicas:

- Esta evaluación contiene 9 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
- La evaluación es **individual**. Un nivel alto de **similitud** con otros estudiantes o fuentes externas no será aceptada y se **anulará** el ejercicio.
- Se puede **consultar material de clase** y utilizar funciones o partes de código desarrollados en clase. Esto ultimo no descontará puntos, pero se debe **hacer referencia a ellos en la entrega**.

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)

- 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:

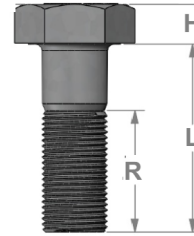
Tabla de puntos (sólo para uso del professor)

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

1. (6 puntos) **Evalúa estructuras de control**

Un perno estructural de cabeza hexagonal, presenta las siguientes dimensiones:

- Altura de cabeza (H) = 1.0 cm
- Largo total (L) = 7.0 cm
- Largo de rosca (R) = 5.0 cm



Una empresa fabrica estos pernos; pero por inexperiencia de sus operarios presentan errores en el proceso de elaboración. Ante esta situación un ingeniero ha determinado la siguiente medida de error:

$$error = \sqrt{(H - 1.0)^2 + (L - 7.0)^2 + (R - 5.0)^2}$$

y determina que un perno es aceptado para venta si el error es menor a 0.5. Elabore un programa que solicite al usuario una cantidad N de pernos a evaluar; posteriormente solicite las medidas de cada perno e imprima un reporte. Algunos ejemplos de diálogo de este programa serían:

Listing 1: Ejemplo 1

```
Cantidad de pernos: 3

Perno 1
Ingrese H:1.2
Ingrese L:7.1
Ingrese R:5.0
Perno aceptado

Perno 2
Ingrese H:1.3
Ingrese L:7.5
Ingrese R:5.1
Perno no aceptado

Perno 3
Ingrese H:1.0
Ingrese L:7.4
Ingrese R:5.3
Perno no aceptado

Cantidad de pernos para venta:1
Cantidad de pernos a desechar:2
```

Listing 2: Ejemplo 1

```
Cantidad de pernos: 6

Perno 1
Ingrese H:1.0
Ingrese L:7.0
Ingrese R:5.0
Perno aceptado

Perno 2
Ingrese H:0.9
Ingrese L:6.9
Ingrese R:4.9
Perno aceptado

Perno 3
Ingrese H:1.1
Ingrese L:7.1
Ingrese R:4.9
Perno aceptado

Perno 4
Ingrese H:1.2
Ingrese L:7.5
Ingrese R:5.0
Perno no aceptado

Perno 5
Ingrese H:1.3
Ingrese L:7.1
Ingrese R:5.1
Perno aceptado

Perno 6
Ingrese H:1.0
Ingrese L:7.4
Ingrese R:5.3
Perno no aceptado

Cantidad de pernos para venta:4
Cantidad de pernos a desechar:2
```

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
Allgoritmo: Evalúa el diseño del algoritmo, siguiendo buenas prácticas en programación. Asi como la ejecución del mismo	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, pero optimizable. La ejecución es correcta (2pts)	El diseño del algoritmo contiene algunos errores que afectan la ejecución (1pts).	El diseño del algoritmo y la ejecución son incorrectos (0 pts)
Código : Evalúa sintaxis en el código y correcta ejecución (semántica)	No contiene errores sintácticos o de compilación. La ejecución es correcta (2pts)	Existen algunos errores sintácticos, que no afectan directamente el resultado, pero hacen al código optimizable. (1.5pts).	Existen errores sintácticos o de ejecución, que afectan parcialmente el resultado (1pts).	El código tiene errores de sintaxis y de ejecución que no permiten obtener un resultado correcto (0 pts).
Eficiencia: evalua uso de buenas practicas en programación en el diseño del algoritmo y el código de programación, para lograr un nivel de eficiencia adecuado	El código es óptimo y eficiente. De buen performance e interacción con el usuario (1pt)	El codigo es de buen performance durante la ejecución pero optimizable. Pero no afecta el resultado. (0.7pts).	El código no esta optimizado, lo que afecta parcialmente el resultado. (0.3pts).	El codigo no esta optimizado y la ejecución es deficiente (0pts).

2. (7 puntos) **Evalúa Funciones**

- La siguiente fórmula permite aproximar el valor de π :

$$\sqrt{6 \times \sum_{i=1}^{\infty} \frac{1}{i^2}} = \sqrt{6 \times \left(\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \dots \right)}$$

- Se pide que elabore una función recursiva **suma(n)** que tenga como parámetros un entero positivo n ; y calcule :

$$\left(\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{n^2} \right)$$

- Posteriormente elabore una función **error(n)** que utilice **suma(n)** para calcular el valor el error de aproximación:

$$\mathbf{error(n)} = \left| 3.141592 - \sqrt{6 \times \mathbf{suma(n)}} \right|$$

- Luego elabore un programa que solicite un número real e_{max} e imprima todos los valores **error(n)** tales que:

$$\mathbf{error(n)} > e_{max}$$

Algunos ejemplos de diálogo de este programa serían:

Listing 3: Ejemplo 1

```
Ingrese emax:0.3
Para i: 1 el error es: 0.692102
Para i: 2 el error es: 0.402979
```

Listing 4: Ejemplo 2

```
Ingrese emax:0.2
Para i: 1 el error es: 0.692102
Para i: 2 el error es: 0.402979
Para i: 3 el error es: 0.283854
Para i: 4 el error es: 0.218979
```

Listing 5: Ejemplo 3

```
Ingrese emax:0.1
Para i: 1 el error es: 0.692102
Para i: 2 el error es: 0.402979
Para i: 3 el error es: 0.283854
Para i: 4 el error es: 0.218979
Para i: 5 el error es: 0.178204
Para i: 6 el error es: 0.150215
Para i: 7 el error es: 0.129818
Para i: 8 el error es: 0.114294
Para i: 9 el error es: 0.102084
```

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado. La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Allgoritmo: Evalúa el diseño del algoritmo, siguiendo buenas prácticas en programación. Asi como la ejecución del mismo	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, pero optimizable. La ejecución es correcta (2pts)	El diseño del algoritmo contiene algunos errores que afectan la ejecución (1pts).	El diseño del algoritmo y la ejecución son incorrectos (0 pts)
Código : Evalúa sintaxis en el código y correcta ejecución (semántica)	No contiene errores sintácticos o de compilación. La ejecución es correcta (2pts)	Existen algunos errores sintácticos, que no afectan directamente el resultado, pero hacen al código optimizable. (1.5pts).	Existen errores sintácticos o de ejecución, que afectan parcialmente el resultado (1pts).	El código tiene errores de sintaxis y de ejecución que no permiten obtener un resultado correcto (0 pts).
Eficiencia: evalua uso de buenas practicas en programación en el diseño del algoritmo y el código de programación, para lograr un nivel de eficiencia adecuado	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2 pts)	El codigo es de buen performance durante la ejecución pero optimizable. Pero no afecta el resultado. (1.5 pts).	El código no esta optimizado, lo que afecta parcialmente el resultado. (0.5pts).	El codigo no esta optimizado y la ejecución es deficiente (0pts).

3. (7 puntos) **Evalúa Arreglos estáticos y punteros**

Las cadenas de ADN estan conformadas por 4 bases nitrogenadas: adenina (A), timina (T), guanina (G) y citosina (C). El proceso mediante el cual el ADN envía un mensaje al citoplasma se denomina transcripción. En el proceso de transcripción, las cadenas de ADN generan una copia de estas mediante los siguientes pasos:

- Las bases adeninas generan timinas,
- las timina generan adeninas,
- las guanina generan citosina,
- las citosina generan guanina.

Por ejemplo:

- Una cadena de ADN

ACTGAC

genera la copia

TGACTG

Se solicita que escriba un programa que genere un arreglo aleatorio de ADN de 20 bases nitrogenadas, luego elabore una función que solicite el arreglo y lo transforme a su copia. Algunos ejemplos de diálogo de este programa serían:

Listing 6: Ejemplo 1

```
Arreglo
T C G A T A G G G A T T T C T C C G C A
Arreglo copia
A G C T A T C C C T A A A G A G G C G T
```

Listing 7: Ejemplo 2

```
Arreglo
A G T A G G A A T C C C G T T T T T G T
Arreglo copia
T C A T C C T T A G G G C A A A A A C A
```

Listing 8: Ejemplo 3

```
Arreglo
C A A A A T C C G A A G A A G G G A T G
Arreglo copia
G T T T T A G G C T T C T T C C C T A C
```


Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado. La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Allgoritmo: Evalúa el diseño del algoritmo, siguiendo buenas prácticas en programación. Asi como la ejecución del mismo	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, pero optimizable. La ejecución es correcta (2pts)	El diseño del algoritmo contiene algunos errores que afectan la ejecución (1pts).	El diseño del algoritmo y la ejecución son incorrectos (0 pts)
Código : Evalúa sintaxis en el código y correcta ejecución (semántica)	No contiene errores sintácticos o de compilación. La ejecución es correcta (2pts)	Existen algunos errores sintácticos, que no afectan directamente el resultado, pero hacen al código optimizable. (1.5pts).	Existen errores sintácticos o de ejecución, que afectan parcialmente el resultado (1pts).	El código tiene errores de sintaxis y de ejecución que no permiten obtener un resultado correcto (0 pts).
Eficiencia: evalua uso de buenas practicas en programación en el diseño del algoritmo y el código de programación, para lograr un nivel de eficiencia adecuado	El código es óptimo y eficiente. De buen performance e interacción con el usuario (2 pts)	El codigo es de buen performance durante la ejecución pero optimizable. Pero no afecta el resultado. (1.5 pts).	El código no esta optimizado, lo que afecta parcialmente el resultado. (0.5pts).	El codigo no esta optimizado y la ejecución es deficiente (0pts).