

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.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	7	
2	6	
3	7	
Total:	20	

1. (7 points) **Números Pitagóricos**

Implemente un programa que encuentre todos los números pitagóricos menores a 30. El programa debe de mostrar en cada linea la combinación de números enteros positivos que cumpla la igualdad:

$$a^2 + b^2 = c^2.$$

Un ejemplo del diálogo de este programa sería:

Listing 1: Ejemplo 1

```
Los numeros pitagorigos menores a 30 son:
3,4,5 => 3**2 + 4**2 = 5**2
6,8,10 => 6**2 + 8**2 = 10**2
.
.
.
```

La rúbrica para esta pregunta es:

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)	La ejecución es correcta (1pts).	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 codigo es de buen performance durante la ejecución (1.5pts).	El codigo no está optimizado pero la ejecución no es deficiente(1pts).	El codigo no está optimizado y la ejecución es deficiente (0pts).

2. (6 points) **Escalera Alternada**

Generar una escalera alternando los símbolos '+' y '*'. El programa debe cumplir las siguientes propiedades:

- El programa debe solicitar el número de caracteres de la base. Este número debe de ser mayor a 1 y menor a 10.
- Empezar cada fila con el símbolo '+'.

Algunos ejemplos de diálogo de este programa serían:

Listing 2: Ejemplo 1

```
Ingrese el numero de caracteres para la base: 0
Ingrese el numero de caracteres para la base: 1
Ingrese el numero de caracteres para la base: 3
+
++
+++
```

Listing 3: Ejemplo 1

```
Ingrese el numero de caracteres para la base: 7
+
++
+++
++++
+++++
++++++
+++++++
```

Listing 4: Ejemplo 1

```
Ingrese el numero de caracteres para la base: 10
Ingrese el numero de caracteres para la base: 9
+
++
+++
++++
+++++
++++++
+++++++
+++++++
+++++++
+++++++
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (2pts)	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (1.5pts)	La ejecución es correcta (1pts).	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) Rodar Números

Utilizando punteros, implemente una función que modifique el orden de una secuencia de números. Para ello considere lo siguiente:

- Crear un array estático con los números del 1 al 6.
- Crear una función para imprimir los elementos del array en una línea y separados por un espacio.
- Crear una función que modifique la posición de los elementos del array original según el número entero ingresado.
 - Si el valor es positivo, trasladar los elementos hacia la izquierda.
 - Si el valor es negativo, trasladar los elementos hacia la derecha.

Algunos ejemplos de diálogo de este programa serían:

Listing 5: Ejemplo 1

```
Array original: 1 2 3 4 5 6
Ingrese un numero: 3
Array modificado: 4 5 6 1 2 3
```

Listing 6: Ejemplo 1

```
Array original: 1 2 3 4 5 6
Ingrese un numero: 0
Array modificado: 1 2 3 4 5 6
```

Listing 7: Ejemplo 1

```
Array original: 1 2 3 4 5 6
Ingrese un numero: -2
Array modificado: 5 6 1 2 3 4
```

La rúbrica para esta pregunta es:

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)	La ejecución es correcta (1pts).	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).