

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) **Ejercicio 1. Estructuras selectivas y repetitivas.**

Un profesor quiere registrar las notas de los alumnos de una practica calificada. Para ello, ingresará la cantida de notas a registrar. Se debe verificar que hayan por lo menos 5 alumnos. También se debe verificar que las notas ingresadas estén entre 0 y 20. Además, dependiendo del valor de la nota, se va a imprimir un mensaje: si la nota está entre 0 y 10, se imprimirá "Este alumno reprobó", si la nota es mayor que 10 y menor o igual a 15, se imprime "El alumno lo hizo bien". Si la nota es mayor a 16 se imprime "El alumno lo hizo muy bien!".

Algunos ejemplos de ejecución del código son:

Listing 1: Ejemplo 1

```
Ingrese el numero de alumnos:5
Ingrese una nota:5
Este alumno reprobó
Ingrese una nota:10
Este alumno reprobó
Ingrese una nota:13
El alumno lo hizo bien
Ingrese una nota:-1
Ingrese una nota:2
Este alumno reprobó
Ingrese una nota:20
El alumno lo hizo muy bien!

Process finished with exit code 0
```

Listing 2: Ejemplo 1

```
Ingrese el numero de alumnos:-1
Ingrese el numero de alumnos:0
Ingrese el numero de alumnos:3
Ingrese el numero de alumnos:5
Ingrese una nota:20
El alumno lo hizo muy bien!
Ingrese una nota:20
El alumno lo hizo muy bien!
Ingrese una nota:20
El alumno lo hizo muy bien!
Ingrese una nota:20
El alumno lo hizo muy bien!
Ingrese una nota:20
El alumno lo hizo muy bien!
```

```
Process finished with exit code 0
```

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).

2. (6 points) Ejercicio 2. Funciones y recursividad.

Escribe una función recursiva `sumarDigitos` que reciba un número entero y devuelva la suma de los dígitos del número.

Algunos ejemplos de la ejecución correcta del código son:

Listing 3: Ejemplo 1

```
sumarDigitos(0)
La suma es: 0
sumarDigitos(1111)
La suma es: 4
sumarDigitos(12345)
La suma es: 15
```

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) Ejercicio 3. Funciones y punteros.

Escribe tres funciones que actúen de la siguiente manera:

- La primera, `recibirDatos`, pide dos números enteros, `n1` y `n2`, al usuario. Los números no deben ser mayores a 7. Luego, crea dos punteros a las variables y los pasa como parámetros a la segunda función.
- La segunda, `imprimirValores`, llama a la tercera función con los mismos parámetros que recibió de la primera, y luego imprime la dirección de los parámetros que recibió, junto con el valor de estos.
- La tercera, `calcularValores`, reemplaza en el valor del puntero a `n1` por $n1 * n2$ y el valor del puntero a `n2` por $n1$ elevado a $n2$.

Algunos ejemplos de la ejecución correcta del código son:

Listing 4: Ejemplo 1

```
Ingrese n1:2
Ingrese n2:3
La direccion de n1 es: 0x7ffc724291e0
El nuevo valor de n1 es: 6
La direccion de n2 es: 0x7ffc724291e4
El nuevo valor de n2 es: 8

Process finished with exit code 0
```

Listing 5: Ejemplo 1

```
Ingrese n1:8
Ingrese n1:7
Ingrese n2:8
Ingrese n2:2
La direccion de n1 es: 0x7ffe5649c7a0
El nuevo valor de n1 es: 14
La direccion de n2 es: 0x7ffe5649c7a4
El nuevo valor de n2 es: 49

Process finished with exit code 0
```

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).