

Indicaciones específicas:

- Esta evaluación contiene 7 páginas (incluyendo esta página) con 5 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.
 - 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	4	
2	4	
3	4	
4	4	
5	4	
Total:	20	

1. (4 points) **Diseño de Patrones**

Implementar una modificación del patrón Singleton en C++:

- La clase Singleton debe permitir crear una cantidad **determinada** de instancias.
- El límite del número de instancias a crear debería de ser un atributo de la clase. Considere para su prueba 3 instancias.
- El constructor debe asignar un valor aleatorio a un atributo de tipo entero llamado **data**. Su valor debería ser un entero entre 0 y 99.
- Si en la función `main()` se crea una instancia más del límite de instancias, entonces esta será igual a la primera. La siguiente instancia será igual a la segunda creada, y del mismo modo con la tercera instancia. La idea es que no se creen más de tres instancias.

Comprobar su implementación con el siguiente código.

Listing 1: Ejemplo 1

```
int main() {
    Singleton *ptr1 = Singleton::getInstance();
    cout << "ptr1: data=" << ptr1->getData() << endl;

    Singleton *ptr2 = Singleton::getInstance();
    cout << "ptr2: data=" << ptr2->getData() << endl;

    Singleton *ptr3 = Singleton::getInstance();
    cout << "ptr3: data=" << ptr3->getData() << endl;

    Singleton *ptr4 = Singleton::getInstance();
    cout << "ptr4: data=" << ptr4->getData() << endl;

    Singleton *ptr5 = Singleton::getInstance();
    cout << "ptr5: data=" << ptr5->getData() << endl;
    return 0;
}

/*
Resultado:

ptr1: data = 83
ptr2: data = 86
ptr3: data = 77
ptr4: data = 83
ptr5: data = 86
*/
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Diseño de Patrones	Selección adecuada de los patrones que permiten la solución de algún problema de diseño con una adecuada justificación y descripción a través de un diagrama de clases. (4pts)	Selección de un patrón que permite la solución de alguno de los problemas de diseño con una justificación y descripción a través de un diagrama de clases. (3pts)	Selección de un patrón que permite la solución de alguno de los problemas de diseño pero no se tiene una justificación y descripción inadecuada del diagrama de clases. (2pts).	Selección de un patrón pero no se justificó y descripción incorrecta del diagrama de clases, (1pts)

2. (4 points) **Pilas y Colas**

(2 pts) En una **Pila** se ingresan, de izquierda a derecha, los siguientes elementos:

Listing 2: Ejemplo 2

E A S * Y * Q U E * * * S T * * * I O * N * * *

Cada letra indica la operación **insertar**, mientras que el asterisco la operación **extraer**. Escriba la secuencia, de izquierda a derecha, de elementos que retornará las operaciones **extraer**. Si la operación **extraer** se ejecuta con la Pila vacía, retorna **EMPTY**.

(2 pts) En una **Cola** se ingresan, de izquierda a derecha, los siguientes elementos:

Listing 3: Ejemplo 3

E A S * Y * Q U E * * * S T * * * I O * N * * *

Cada letra indica la operación **insertar**, mientras que el asterisco la operación **extraer**. Escriba la secuencia, de izquierda a derecha, de elementos que retornará las operaciones **extraer**. Si la operación **extraer** se ejecuta con la Cola vacía, retorna **EMPTY**.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Implementación del Algoritmo	La implementación del algoritmo es clara y ordenada. El desarrollo de la solución es detallado y correcto (4pts).	La implementación del algoritmo es clara y ordenada. El desarrollo de la solución es suficiente (3pts)	La solución es parcialmente correcta (2pts).	La solución es incorrecta(1pts)

3. (4 points) **Árboles Binarios de Búsqueda**

Dibujar el árbol binario de búsqueda si se insertaron los nodos en el siguiente orden:

Listing 4: Algoritmo 1

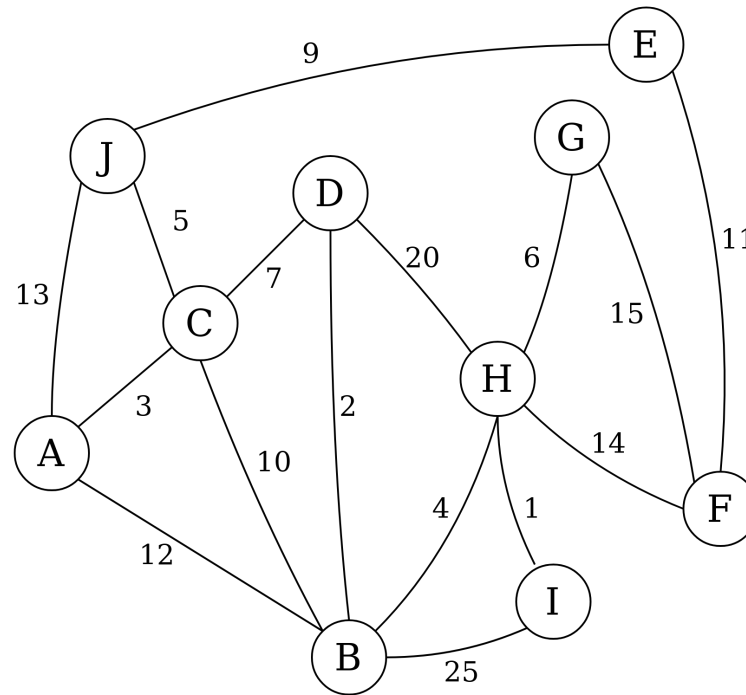
```
arbol.insert(25)
arbol.insert(20)
arbol.insert(36)
arbol.insert(10)
arbol.insert(22)
arbol.insert(30)
arbol.insert(40)
arbol.insert(5)
arbol.insert(12)
arbol.insert(28)
arbol.insert(38)
arbol.insert(48)
arbol.insert(2)
arbol.insert(7)
arbol.insert(16)
arbol.insert(26)
arbol.insert(44)
arbol.insert(51)
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Implementación del Algoritmo	La implementación del algoritmo es clara y ordenada. El desarrollo de la solución es detallado y correcto (4pts).	La implementación del algoritmo es clara y ordenada. El desarrollo de la solución es suficiente (3pts)	La solución es parcialmente correcta (2pts).	La solución es incorrecta(1pts)

4. (4 points) **Álgoritmo de Kruskal**

Identificar el Árbol de Expansión Mínimo en el siguiente grafo:

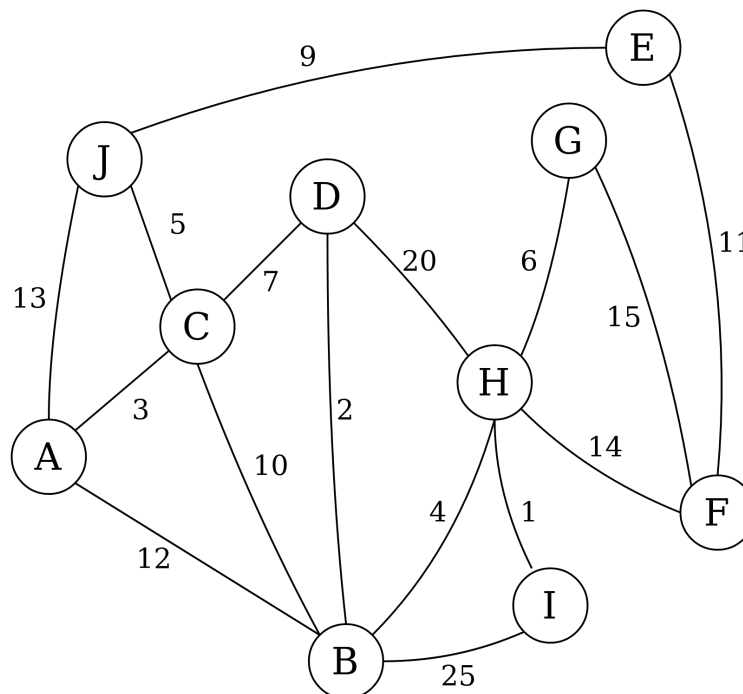


La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Implementación del Álgortmo	La imple- mentación del algoritmo es clara y or- denada. El desarrollo de la solución es detallado y cor- recto (4pts).	La imple- mentación del algoritmo es clara y or- denada. El desarrollo de la solución es suficiente (3pts)	La solución es parcialmente correcto (2pts).	La solución es incorrecta(1pts)

5. (4 points) **Álgoritmo de Dijkstra**

En el siguiente grafo, identificar el camino con menor costo desde el vértice A (origen) a todos los demás vértice.



Utilice el algoritmo de Dijkstra.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Implementación del Álgortmo	La imple- mentación del algoritmo es clara y or- denada. El desarrollo de la solución es detallado y cor- recto (4pts).	La imple- mentación del algoritmo clara y ordenada. El desarrollo de la solución es suficiente (3pts)	La solución es parcialmente correcta (2pts).	La solución es incorrecta(1pts)