

Indicaciones específicas:

- Esta evaluación contiene 9 páginas (incluyendo esta página) con 4 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
 - p4.cpp
- 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	5	
2	5	
3	5	
4	5	
Total:	20	

1. (5 points) **Pregunta 1**

En Criptografía, el *Cifrado por Desplazamiento* permite esconder un mensaje utilizando la siguiente función:

$$c = F(x) = (ax + b) \% 26,$$

donde x es una letra representada por un número entero entre 0 y 25, 0 para la letra A y 25 para Z (sin usar la Ñ). Note que la función F tiene los parámetros a y b , los cuales deben ser asignados al crear la función. Dicho esto:

- Implemente un Functor para cifrar, con $a = 5$ y $b = 6$, la frase PROGRAMACION.
- Luego implemente otro Functor para descifrar, con $a = 21$ y $b = 4$, la frase PYJGOCTVYATQOGSOGS.

El ejercicio debe ser validado con el siguiente bloque de código:

```
int main(){
    Cipher F(5,6); // Functor para cifrar, a=5 y b=6
    cout << F('Y') << endl; // Resultado: W

    Cipher H(21,4); // Functor para descifrar, a=21 y b=4
    cout << H(F('Y')) << endl; // Resultado: Y

    string s = "PROGRAMACION";
    cout << "Cifrando...PROGRAMACION" << endl;
    // Imprima el cifrado de 's'

    string u = "PYJGOCTVYATQOGSOGS";
    cout << "Descifrando...PYJGOCTVYATQOGSOGS" << endl;
    // Imprima el descifrado de 'u'

}
```

La rúbrica para esta pregunta es:

Criterio		Excelente	Adecuado	Mínimo	Insuficiente
Librería Es- tandar		Selección del contenedor de acuerdo con lo solicitado, uso adecuado de los iteradores, estructuras genéricas basados en contenedores. (5pts)	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. (1pts)

2. (5 points) **Pregunta 2**

Para el siguiente ejercicio:

- ¿Que proceso realiza la función `foo` en el vector `V`?, mencione su hipótesis a modo de comentario.
- Encuentre un Invariante de Bucle que lo ayude a demostrar su hipótesis. Imprima en cada iteración el Invariante de Bucle para su validación.
- ¿Que sucede al terminar el bucle?, demuestre su hipótesis utilizando el Invariante de Bucle.
- Encuentre la complejidad algorítmica de `foo`.

A continuación se muestra la función `foo`:

```
template<typename T>
void foo(vector<T>& V){
    int n  = V.size();
    int i  = (n - 1) / 2;
    int j  = n / 2;
    T tmp;
    while (i >= 0 && j <= (n-1)){
        tmp = V[i];
        V[i] = V[j];
        V[j] = tmp;
        i--;
        j++;
    }
}
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Complejidad Algorítmica	Buen nivel de abstracción, el problema logro realizar con la complejidad solicitada, funciona correctamente y sin errores. (5pts)	Buen nivel de abstracción, el problema logro realizar lo solicitado sin lograr alcanzar la complejidad solicitada, funciona correctamente y sin errores. (4pts)	Programa no funciona adecuadamente, bajo nivel de abstracción, más de 3 errores, nivel de complejidad algorítmica incorrecta. (2pts)	Se intento pero no se logró que funcione lo solicitado. (1pts)

3. (5 points) **Pregunta 3**

El propósito del siguiente programa es adaptar el método **draw** de la clase **Linea** a una que el usuario pueda utilizar directamente (**draw_figure**). Completar el siguiente bloque de código:

```
struct Point {
    // Completar
};

void draw_figure(Point &p) {
    // Completar
}

struct Line {
    Point _inicio;
    Point _final;
    Line(int x1, int y1, int x2, int y2){
        _inicio.x = x1;
        _inicio.y = y1;
        _final.x = x2;
        _final.y = y2;
    }
    void draw(){
        cout <<"Linea:␣";
        cout << _inicio;
        cout << "-----";
        cout << _final;
        cout << endl;
    }
};

int main() {
    Line obj1(2,2,5,6);
    LineAdapter adapter(obj1);
    draw_figure(adapter);

    // Resultado:
    // Linea: P(2,2)-----P(5,6)

    return 0;
}
```

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. (5pts)	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. (4pts)	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)

4. (5 points) **Pregunta 4**

El siguiente programa calcula, de forma secuencial, la media y desviación estándar de una secuencia de números enteros. Utilizando programación concurrente acelere el cálculo de estos dos valores, para ello:

- Genere un vector de 10 millones de números.
- Utilice la librería `chrono` para medir los tiempos.
- Compare su tiempo con el del programa original e indique con cuantos threads obtiene el tiempo más corto.

```
void random_ints(vector<int>& v, int N){
    for(int i=0; i<N; i++)
        v.push_back(rand() % 200 - 100);
}

template<typename T>
void info(vector<T> V, double& mean, double& std){
    for(const T& item: V)
        mean += item;
    mean = mean / V.size();

    for(const T& item: V)
        std += pow(item - mean, 2);
    std = std / V.size();
    std = sqrt(std);
}

int main(){
    vector<int> vec;
    random_ints(vec, 10000000);

    double mean = 0.;
    double std = 0.;
    info(vec, mean, std);

    cout << "Media:_" << mean << endl;
    cout << "Desviacion_Estandar:_" << std << endl;
}
```

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. (5pts)	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. (4pts)	Programa no funciona, bajo nivel de abstracción, más de 3 errores visibles , no se usa los hilos adecuadamente ni un control de race condition. (2pts)	Contiene errores que no hace que funcione el programa. (1pts)