

2. Problema de Algoritmia (5 puntos - 15 minutos)

<pre> long long int sucesion(int n){ if(n==0) return 1; if(n==1) return 2; long long int r1=sucesion(n-1); long long int r2=sucesion(n-2); return 2*r1-3*r2; } </pre>	<pre> long long int sucesion_m(int n){ if(n==0) return 1; if(n==1) return 2; long long int r1, r2; if(memoi[n-1]>0) r1=memoi[n-1]; else{ r1=sucesion_m(n-1); memoi[n-1]=r1; } if(memoi[n-2]>0) r2=memoi[n-2]; else{ r2=sucesion_m(n-2); memoi[n-2]=r2; } return 2*r1-3*r2; } </pre>
<pre> #include <iostream> #include <vector> #include <algorithm> #include <iterator> #define N 4 int main(){ for(int n=0; n<N; n++){ cout<<"["<n<<"<<":"<<\ sucesion(n)<<endl; } } </pre> <p style="text-align: center;">A</p>	<pre> #include <iostream> #include <vector> #include <algorithm> #include <iterator> #define N 4 vector<long long int> memoi(N); int main(){ for(int n=0; n<N; n++){ memoi.clear(); //inicialización de memoi *** cout<<"["<n<<"<<":"<<\ sucesion_m(n)<<endl; } //muestra memoi en pantalla copy(memoi.begin(), ***, ***); } </pre> <p style="text-align: center;">B</p>

La primera fila de la tabla superior implementa dos sucesiones—*sucesion* y *sucesion_m*— de manera recursiva. La segunda sucesión emplea un vector global auxiliar *memoi*. La segunda fila de la tabla contiene dos códigos cliente: A-invoca a *sucesion* y B-invoca a *sucesion_m*.

Se pide:

- Expresión recursiva de ambas sucesiones (2 puntos)

$$a_n = 2a_{n-1} - 3a_{n-2} \quad \text{en ambos casos}$$

2. Salida en pantalla para el caso A (2 puntos)

```
[0]:1
[1]:2
[2]:1
[3]:-4
```

3. Razone si la complejidad computacional es polinomial o exponencial (en función de N) para los casos A y B. Justifique brevemente la respuesta (2 puntos).

Caso A: Exponencial: árbol binario

Caso B: Polinomial puesto que la ramificación del árbol de recursión sólo ocurre para valores no computados anteriormente

4. Complete los `***` del código para el caso B de manera conveniente (escriba la respuesta aquí) (2 puntos)

```
for(int j=0; j<N; j++){
    memo1.push_back(-1);
}

copy(memo1.begin(), memo1.end(), ostream_iterator<long long int>(cout, " "));
```

5. Salida en pantalla para el caso B (2 puntos)

```
[0]:1
[1]:2
[2]:1
[3]:-4
```

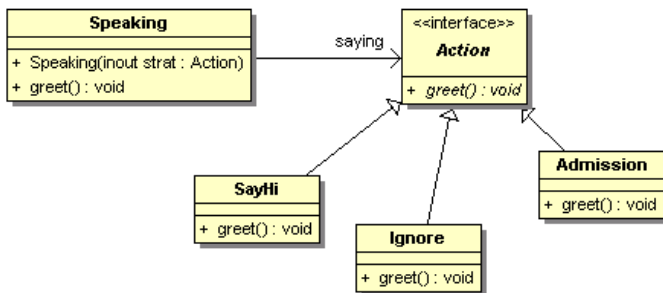
```
1 2 1 -1
```

3. Problema de Análisis y Diseño Orientado a Objetos (10 puntos - 30 minutos)

Para el código que se adjunta, se pide.

1. Ingeniería inversa (2.5 puntos).
2. Nombre del patrón aplicado (1 punto)
3. Resultado de la ejecución del programa en la consola del sistema (1.5 puntos).

1.



2. Estrategia GoF

3.

Hi! How's it going?

<Pretend I don't see you>

I'm sorry. I forgot your name.

```

#include <iostream>
using namespace std;

class Action {
public:
    virtual void greet() = 0;
};

class SayHi : public Action {
public:
    void greet() {
        cout << "Hi! How's it going?" << endl;
    }
};

class Ignore : public Action {
public:
    void greet() {
        cout << "(Pretend I don't see you)" << endl;
    }
};

class Admission : public Action {
public:
    void greet() {
        cout << "I'm sorry. I forgot your name." <<
endl;
    }
};

class Speaking {
    Action& saying;
public:
    Speaking(Action& strat) : saying(strat) {}
    void greet() { saying.greet(); }
};

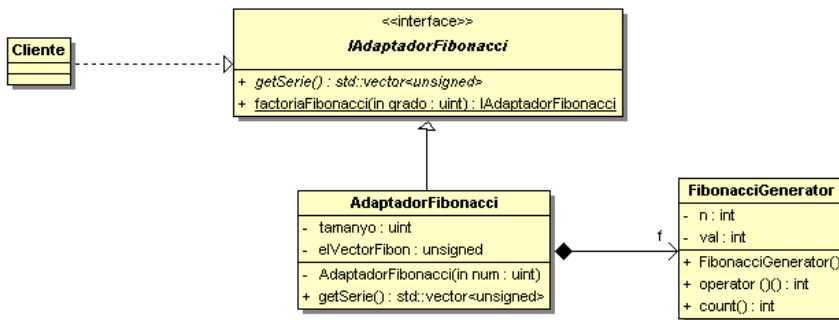
int main() {
    SayHi sayhi;
    Ignore ignore;
    Admission admission;
    Speaking c1(sayhi), c2(ignore), c3(admission);
    c1.greet();
    c2.greet();
    c3.greet();

    return 0;
}
  
```

Para el siguiente código de test y DCD se pide:

```
int main() {
const unsigned numeroFibo = 20;
IAdaptadorFibonacci *pAdaptadorFibo =
    IAdaptadorFibonacci::factoriaFibonacci(numeroFibo);
cout << "Tabla de Fibonacci" <<endl;
cout << "-----" <<endl;
for_each(pAdaptadorFibo->getSerie().begin(),
    pAdaptadorFibo->getSerie().end(), imprimir);
return 0;
}
void imprimir(unsigned numFibo){
    static unsigned indice;
    cout << indice++ << ": " << numFibo <<endl;
}
```

```
#ifndef FIBONACCIGENERATOR_H
#define FIBONACCIGENERATOR_H
class FibonacciGenerator {
    int n;
    int val[2];
public:
    FibonacciGenerator() : n(0) { val[0] = val[1] = 1; }
    int operator()() {
        int result = n > 2 ? val[0] + val[1] : 1;
        ++n;
        val[0] = val[1];
        val[1] = result;
        return result;
    }
    int count() { return n; }
};
#endif // FIBONACCIGENERATOR_H
```



4. Implementación en C++ de las clases *IAdaptadorFibonacci* y *AdaptadorFibonacci* (5 puntos).

```
#include <vector>
#include "FibonacciGenerator.h"
class IAdaptadorFibonacci
{
public:
virtual std::vector<unsigned> & getSerie() = 0;
    static IAdaptadorFibonacci *factoriaFibonacci(unsigned grado) {
        return (new AdaptadorFibonacci(grado));
    }
};
class AdaptadorFibonacci: public IAdaptadorFibonacci
{
    FibonacciGenerator f;
    unsigned tamaño;
    std::vector<unsigned> elVectorFibon;
    friend class IAdaptadorFibonacci;
    AdaptadorFibonacci(unsigned num): tamaño(num) {
        for (unsigned i=0; i<=tamaño; i++)
            elVectorFibon.push_back(f());
    }
public:
    virtual std::vector<unsigned> & getSerie()
    {return elVectorFibon;}
};
```

APELLIDOS

NOMBRE

Nº Mat.

ASIGNATURA: SISTEMAS INFORMÁTICOS INDUSTRIALES

Calificación

CURSO 4º

GRUPO

Enero 2015

5. Problema de Sistemas Operativos (10 puntos - 30 minutos)

Referente al capítulo de Procesos el sistema operativo nos ofrece los siguientes servicios según la interfaz POSIX en lenguaje de programación C;

`fork(), getpid(), wait()`

- 1º- (1 punto) Elija un servicio de los anteriores, escriba el prototipo de la función (parámetros y valor devuelto) y explique para qué sirve.
- 2º- (1 punto) Del servicio elegido codifique en C un breve ejemplo, diciendo qué hace.

Referente al capítulo de Comunicación y Sincronismo el sistema operativo nos ofrece los siguientes servicios según la interfaz POSIX en lenguaje de programación C;

`pipe(), sem_init(), sem_open()`

- 3º- (1 punto) Elija un servicio de los anteriores, escriba el prototipo de la función (parámetros y valor devuelto) y explique para qué sirve.
- 4º- (1 punto) Del servicio elegido codifique un breve ejemplo, diciendo qué hace.

Referente al capítulo de Sistema de Ficheros el sistema operativo nos ofrece los siguientes servicios según la interfaz POSIX en lenguaje de programación C;

`read(), lseek(), open()`

- 5º- (1 punto) Elija un servicio de los anteriores, escriba el prototipo de la función (parámetros y valor devuelto) y explique para qué sirve.
- 6º- (1 punto) Del servicio elegido codifique un breve ejemplo, diciendo qué hace.

Referente al capítulo de Distribuidos el sistema operativo nos ofrece los siguientes servicios según la interfaz POSIX en lenguaje de programación C;

`bind(), listen(), accept()`

- 7º- (1 punto) Elija un servicio de los anteriores, escriba el prototipo de la función (parámetros y valor devuelto) y explique para qué sirve.
- 8º- (1 punto) Del servicio elegido codifique un breve ejemplo, diciendo qué hace.

Referente al capítulo de Gestión de Memoria rellene la tabla siguiente con las características de las dos siguientes regiones de un proceso:

- 9º- (1 punto) Región de variables globales con valor inicial.
- 10º- (1 punto) Región de biblioteca dinámica.

Nombre de la región	Permisos de la región	Procedencia de los datos	¿Se rellena de ceros?	¿Es una región compartida o privada?	Posibles ubicaciones de las páginas de esta región	¿Puede crecer?
Código	R, X	del fichero ejecutable	No	Compartida	Memoria o ejecutable	No
...						