

1.- ¿Qué muestra por pantalla?

```
#include <iostream>

struct Foo{
    int x{0};
    int y{0};
};
struct Fii{
    int x{-1};
    int y{-1};
};

std::ostream& operator<<(std::ostream &o,const Foo &f)
{
    o<<f.y<<f.x<<std::endl;
    return o;
}

std::istream& operator>>(std::istream &i,Foo &f)
{
    i>>f.x>>f.y;
    return i;
}

int main()
{
    Foo fii;

    std::cin>>fii; //Usuario teclea: 1 (pulsa espacio) 2 (pulsa enter)
    std::cout<<fii;

    return 0;
}
```

21

12

-1-1

Error de compilación

2.- ¿Qué muestra por pantalla?

```
#include <iostream>

struct Foo{
    int x{0};
    int y{0};
};

std::ostream& operator<<(std::ostream &o,const Foo &f)
{
    o<<f.x<<f.y;
    return o;
}

Foo operator+(Foo &a,Foo &b)
{
    a.x=a.x+b.x;
    a.y=a.y+b.y;

    return b;
}

int main()
{
    Foo fii{1,2},foo{4,1};

    std::cout<<fii+foo;
    std::cout<<fii;

    return 0;
}
```

4153

5353

Error de compilación

5312

3.-¿Qué muestra por pantalla?

```
#include <iostream>

struct Foo{
    int x{0};
    int y{0};
};

std::ostream& operator<<(std::ostream &o,const Foo &f)
{
    o<<f.x<<f.y;
    return o;
}

Foo operator+(const Foo &a, const Foo &b)
{
    a.x=a.x+b.x;
    a.y=a.y+b.y;

    return b;
}

int main()
{
    Foo fii{1,2},foo{4,1};

    std::cout<<fii+foo;
    std::cout<<fii;

    return 0;
}
```

4153

5353

Error de compilación

5312

Comentado [3]: a es constante en la sobrecarga del operador + => no podemos asignarle ningún valor

4.- ¿Qué muestra por pantalla?

```
#include <iostream>

struct tas{
    tas(){
        x=0;y=1;
    }
    int x{1};
    int y{1};
};

std::ostream & operator <<(std::ostream &os, const tas & t)
{
    os<<t.x+1<<"_"<<t.y+1;
    return os;
}

int main()
{
    tas miTAS;
    std::cout<<miTAS;

    return 0;
}
```

0 1|

1 2

Error de compilación

2_2

Comentado [4]: al hacer `cout<<miTAS` cargamos en el buffer `0+1_1+1`. Que es lo que mostramos por terminal

5.- ¿Qué muestra por pantalla?

```
#include <iostream>

struct tas{
    int x{1};
    int y{1};
};

bool operator >(const tas &t1,const tas &t2)
{
    return (t1.x>t2.x and t1.y<t2.y);
}

int main()
{
    tas miTAS1,miTAS2{2,2};
    if(miTAS2>miTAS1)
    {
        std::cout<<2;
    }
    else
    {
        std::cout<<1;
    }

    return 0;
}
```

```
1
2
10
20
```

6.- ¿Qué muestra por pantalla?

```
#include <iostream>
class Foo{
    int a;
    int b;
public :
    Foo(){a=5;b=7;
        std::cout<<b;};
    void print(){std::cout<<a<<b;};
};

int main()
{
    Foo fi;
    fi.print();
    return 0 ;
}
```

757

57

Error de compilación

570

7.- ¿Qué muestra por pantalla?

```
#include <iostream>

void funcion(int in1,int in2)
{
    try
    {
        try
        {
            try {
                if(in1>in2)
                    {throw true;}
                else
                    {throw false;}
            } catch (int c1) {
                std::cout<<"1";
            }

            }catch(double c2){
                std::cout<<"2";
            }
        }
    } catch(...)
    {
        std::cout<<"3";
    }
}

int main()
{
    funcion(0.2,2);
    return 0;
}
```

1

2

3

Error de compilación

Comentado [7]: Llamamos a función (0.2,2)

funcion recibe dos tipos int => cast implicito

0 no es mayor que 2 => ejecutamos else y lanzamos una excepción de tipo bool

El catch int no nos vale=> vamos hacia atrás

El catch double tampoco nos vale => seguimos hacia atrás

El catch "genérico" (...) si nos vale => mostramos 3

8.- ¿Qué muestra por pantalla? (repetido)

```
#include <iostream>
#include <memory>

using namespace std;

int main()
{
    int k{3};

    auto p=make_shared<int>(k);
    auto p1=make_shared<int>(k);

    cout<<*p1<<*p;

    *p+=1;
    cout<<*p1<<*p;
    *p1+=1;
    cout<<*p1<<*p;

    return 0;
}
```

```
333444
334344
333445
334455
```


9.- ¿Qué muestra por pantalla?

```
#include <iostream>
#include <memory>

using namespace std;

int main()
{
    int k{3};

    auto p=make_shared<int>(k);
    auto p1=&p;

    cout<<*p;
    **p1+=1;
    cout<<*p;

    return 0;
}
```

33

34

Dos direcciones de memoria en hexadecimal
Error de compilación

Comentado [9]: p es un shared_ptr<int>
p1 es la dirección de un shared_ptr<int> => un
shared_ptr<int>*

Mostramos el valor al que apunta p => 3

Incrementamos el valor" al que apunta la dirección a la
que apunta p1" => 4

Atención que k sigue valiendo 3... (solo utilizamos su
valor para inicializar el contenido de la dirección a la
que apunta el puntero)

10.-¿Qué muestra por pantalla? (repetido)

```
#include <iostream>
#include <memory>

using namespace std;

int main()
{
    int k{3};

    auto p=make_shared<int>(k);
    auto p1=p;

    cout<<*p1<<*p;

    *p+=1;
    cout<<*p1<<*p;
    *p1+=1;
    cout<<*p1<<*p;

    return 0;
}
```

```
334455
334344
333445
333444
```