1.- ¿Qué muestra por pantalla?

```
#include <iostream>

struct flip{
   flip(){std::cout<<"1";}
   ~flip(){std::cout<<"2";}
};
struct flop:flip{
   flop(){std::cout<<"3";}
   ~flop(){std::cout<<"4";}
};

int main()
{
   flop miFlop;

   return 0;
}
```

1342
1324
3142
3124

2.- ¿Qué muestra por pantalla?

```
#include <iostream>

struct flip{
  flip(){std::cout<<"4";}
  ~flip(){std::cout<<"3";}
};
struct flop:flip{
  flop(){std::cout<<"2";}
  ~flop(){std::cout<<"1";}
};
void fun(const flop &f)
{
  std::cout<<"5";
}

int main()
{
  flop miFlop;
  fun(miFlop);

  return 0;
}
```

42513
4251313
24531
424253131

3.- ¿Qué muestra por pantalla?

```
#include <iostream>

struct flip{
  flip(){std::cout<<"4";}
  ~flip(){std::cout<<"3";}
};
struct flop:flip{
  flop(){std::cout<<"2";}
  ~flop(){std::cout<<"1";}
};
void fun(flop f)
{
  std::cout<<"5";
}

int main()
{
  flop miFlop;
  fun(miFlop);

  return 0;
}
```

42513
4251313
24531
242453131

4.- ¿Qué muestra por pantalla?

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

struct flip{
   int a{3};
   flip(int in):a{in}{std::cout<<a;}
   ~flip(){std::cout<<"flip";};
};

void fun(std::shared_ptr<flip> &in)
{
   in=std::make_shared<flip>(8);
}

int main()
{
   auto a=std::make_shared<flip>(4);
   std::cout<<a->a;
   fun(a);
   std::cout<<a->a;

   return 0;
}
```

448flip8flip
4488flip
448flip4flip
4484flip

5.- ¿Qué muestra por pantalla?

```
#include <iostream>
using namespace std;

int main()
{
    try{
        try{
            if(true)
            {
                throw string{"e"};
            }
        }
        catch(string e){
            cout << "good" << "-";
        }
        cout << "flip-";
    }
    catch(string &e)
    {
        cout << e << "-";
    }

    cout<<"end";

    return 0;
}
```

e-end
e
good-flip-end
good-flip-e-end

6.- ¿Qué muestra por pantalla?

```
#include <iostream>
#include <memory>
using namespace std;

class Sup{
public:
    Sup(){}
    virtual void tip(){cout<<"tip-";}
};

class Trus: public Sup{
public:
    Trus(){}
    void tip(){cout<<"top-";}
};

int main()
{
    Trus a;
    a.tip();

    Sup b;
    b.tip();

    shared_ptr<Sup> c=make_shared<Trus>(Trus());
    c->tip();

    shared_ptr<Trus> d=make_shared<Trus>(Trus());
    d->tip();
}
```

top-tip-top-tip-

top-tip-top-top-

tip-tip-top-top-

top-tip-tip-tip-

7.- ¿Qué muestra por pantalla?

```
#include <iostream>
using namespace std;

struct A{
    int a;
};

int operator==(A const &a, A const &b)
{
    return a.a!=b.a;
}

int main()
{
    A a{3};
    A b{3};

    if(a==b)
    {
        cout<<"super";
    }
    else
    {
        cout<<"vaya";
    }
}
```

Error de compilación
super
[✓ vaya]
supervaya

8.- ¿Qué muestra por pantalla?

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

int main()
{
  auto a=std::make_shared<int>(3.2);
  auto b=*a;
  std::cout<<b;
  b=5;
  std::cout<<*a<<b;

  return 0;
}
```

Error de compilación
[✓ 335]
0x…35
3.235

9.- ¿Qué muestra por pantalla?

```
#include <iostream>

void flip(char a)
{std::cout<<"1";}

void flip(std::string a)
{std::cout<<"2";}

void flip(int a)
{std::cout<<"3";}

int main()
{
   flip('1');
   flip("1");
   auto b=1;
   flip(b);

   return 0;
}
```

123
113
212
112

10.- ¿Qué líneas presentan algún error de compilación?

1	#include <iostream>
2	class **flip**{
3	public:
4	**flip**(int **a**,int **b**, int **c**){pub=a;pri=b;pro=c;}
5	int **pub**;
6	private:
7	int **pri**;
8	protected:
9	int **pro**;
10	};
11	class **flop**:public flip{
12	**flop**():flip(){}
13	int **o**{8};
14	public:
15	**flop**(int **a**,int **b**,int **c**,int **d**):flip(a,b,c),o{6}{}
16	void **resetear**(){
17	pro=0;
18	pri=0;
19	pub=0;
20	}
21	};
22	int **main**(){
23	flop **f**;
24	std::cout<<f.pub;
25	std::cout<<f.o;
26	}
	12; 18; 23; 25 12; 15; 17; 18 15; 18; 23 17; 18; 25