

## ЗАДАЧА

Решение задачи можно присылать (предпочтительнее ссылка на github, и т.п.) на [vkonovodov@gmail.com](mailto:vkonovodov@gmail.com). Бонусные баллы за решения задачи могут быть поставлены только нескольким первым приславшим правильное решение. Решение должно быть оформлено в виде компилирующегося кода.

В следующей программе не вызываются перегруженные `operator new` и `operator delete`.

```
#include <iostream>

class A {
public:
    // ...

    static void * operator new(size_t size) {
        std::cout << "operator new!" << std::endl;
        return ::operator new(size);
    }

    static void operator delete(void *p, size_t size) {
        std::cout << "operator delete!" << std::endl;
        return ::operator delete(p);
    }
};

int main() {
    auto sp = std::make_shared<A>();
}
```

Используя функцию `std::allocate_shared` и свой аллокатор, реализуйте создание `std::shared_ptr<A> sp`, при котором вызывается перегруженный `operator new`, но при этом не используется конструкция `new A` в создающем объект `sp` коде.

**Важно!** Реализованный вами кастомный шаблонный аллокатор не должен использовать явное обращение к классу `A` вида `A::operator new`. Класс `A` должен быть инстанцированием шаблона.