



Was wird ausgegeben?

```
#include<iostream>
#include<string>

struct MyString {
    std::string* str;
    MyString (const std::string& s){
        str = new std::string (s);
    }
    void change (int index, char c){
        (*str)[index] = c;
    }
};

int main(){
    MyString s1("Ameise");
    MyString s2 = s1;
    s1.change(0, 'B');
    std::cout << *(s1.str) << "\n" << *(s2.str);
    return 0;
}
```

- 1 Ameise
Bmeise
- 2 Ameise
Ameise
- 3 Bmeise
Ameise
- 4 Bmeise
Bmeise
- 5 Compiler
Fehler



Was wird ausgegeben?

```
#include<iostream>
#include<string>

struct MyString {
    std::string* str;
    MyString (const std::string& s){
        str = new std::string (s);
    }
    void change (int index, char c){
        (*str)[index] = c;
    }
};

int main(){
    MyString s1("Ameise");
    MyString s2 = s1;
    s1.change(0, 'B');
    std::cout << *(s1.str) << "\n" << *(s2.str);
    return 0;
}
```

- 1 Ameise
Bmeise
- 2 Ameise
Ameise
- 3 Bmeise
Ameise
- 4 Bmeise
Bmeise
- 5 Compiler
Fehler



Was wird ausgegeben?

```
#include<iostream>
#include<string>

struct MyString {
    std::string* str;
    MyString (const std::string& s){
        str = new std::string (s);
    }
    void change (int index, char c){
        (*str)[index] = c;
    }
};

int main(){
    MyString s1("Ameise");
    MyString s2 = s1;
    s1.change(0, 'B');
    std::cout << *(s1.str) << "\n" << *(s2.str);
    return 0;
}
```

1 Ameise
Bmeise

2 Ameise
Ameise

3 Bmeise
Ameise

4 Bmeise
Bmeise

5 Compiler
Fehler





Was wird ausgegeben?

```
#include<iostream>
#include<string>

struct MyString {
    std::string* str;
    MyString (const std::string& s){ // not the
        str = new std::string (s); // copy
    } // constructor!
    void change (int index, char c){
        (*str)[index] = c;
    }
};

int main(){
    MyString s1("Ameise");
    MyString s2 = s1;
    s1.change(0, 'B');
    std::cout << *(s1.str) << "\n" << *(s2.str);
    return 0;
}
```

1 Ameise
Bmeise

2 Ameise
Ameise

3 Bmeise
Ameise

4 Bmeise
Bmeise ●

5 Compiler
Fehler