

```

#include <iostream>
using namespace std;

int wired(int, int);

int main()
{
    int a=4, b=5, c;

    c = wired(a,b);
    cout << a << " " << b << " " << c << endl;

    return 0;
}

int wired(int a, int b)
{
    a = a * 3;
    b = b * 4;
    return (a+b);
}

```

CALL BY VALUE

Argumente
bleiben
unverändert

Kopien von a
und b werden
verändert

„Call-By-Value“ bezeichnet die Übergabe von *Argumentwerten* an eine Funktion, was bedeutet, dass eine solche Funktion mit *Kopien* der übergebenen Argumente arbeitet und nicht mit den Originalvariablen.

```

#include <iostream>
using namespace std;

int wired(int&, int&);

int main()
{
    int a=4, b=5, c;

    c = wired(a,b);
    cout << a << " " << b << " " << c << endl;

    return 0;
}

int wired(int &a, int &b)
{
    a = a * 3;
    b = b * 4;
    return (a+b);
}

```

CALL BY REFERENCE

Argumente
werden
verändert

übergebene
Argumente
werden direkt
verändert

„Call-By-Reference“ bezeichnet die Übergabe von *Argumentadressen* an eine Funktion, was bedeutet, dass eine solche Funktion mit den *Originalen* der übergebenen Argumente arbeitet.