

# Klausurvorbereitung

**STL**

## Wie ist die Ausgabe auf den Bildschirm

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

list<int> cont;
for (int i=1; i<3; ++i) {
    cont.push_back(i);
}

// Alle Elemente über Iteratoren
// ausgeben:
list<int>::iterator iter = cont.begin();

for (; iter != cont.end(); ++iter) {
    cout << *iter << ' ';
}
```

Ausgabe ?

1. 0 1
2. 1 2 1 2
3. 1 2
4. 1 1

## Wie ist die Ausgabe auf den Bildschirm

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

list<int> cont;
for (int i=1; i<3; ++i) {
    cont.push_back(i);
}

// Alle Elemente über Iteratoren
// ausgeben:
list<int>::iterator iter = cont.begin();

for (; iter != cont.end(); ++iter) {
    cout << *iter << ' ';
}
```

Ausgabe ?

```
1. 0 1
2. 1 2 1 2
3. 1 2
4. 1 1
```

Ergebnis

```
__ 0 1      __ 1 2 1 2
_- 1 2      __ 1 1
```

## BildschirmAusgabe?

```
void Print(int j) {  
    cout << j << " ";  
}  
int main() {  
    list<int> cont;  
    cont.push_back(4);  
    cont.push_back(9);  
    cont.push_back(3);  
    cont.push_back(8);  
    cont.push_back(7);  
    for_each(cont.begin(), cont.end(), Print);  
}
```

Ausgabe ?

1. 4 9 3 8 7
2. 1 2 3
3. 1 2 3 4 5
4. Keine Ausgabe

## BildschirmAusgabe?

```
void Print(int j) {  
    cout << j << " ";  
}  
int main() {  
    list<int> cont;  
    cont.push_back(4);  
    cont.push_back(9);  
    cont.push_back(3);  
    cont.push_back(8);  
    cont.push_back(7);  
    for_each(cont.begin(), cont.end(), Print);  
}
```

Ausgabe ?

1. 4 9 3 8 7
2. 1 2 3
3. 1 2 3 4 5
4. Keine Ausgabe

Ergebnis

\_\_ - 4 9 3 8 7      \_\_\_ 1 2 3  
\_\_\_ 1 2 3 4 5      \_\_\_ keine

## BildschirmAusgabe? Letztes Print

```
void Print(int j) {  
    cout << j << " ";  
}  
int main() {  
    list<int> cont;  
    cont.push_back(4); cont.push_back(9);  
    cont.push_back(3); cont.push_back(8);  
    cont.push_back(7);  
    // Ausgabe ist 4 9 3 8 7  
    for_each(cont.begin(), cont.end(), Print);  
  
    list<int>::iterator iter =  
        find(cont.begin(), cont.end(), 3);  
    for_each(iter, cont.end(), Print);  
}
```

Ausgabe ?

```
1. 4 9 3 8 7  
2. 3 8 7  
3. 3  
4. 3 4 7 8 9
```

## BildschirmAusgabe? Letztes Print

```
void Print(int j) {  
    cout << j << " ";  
}  
int main() {  
    list<int> cont;  
    cont.push_back(4); cont.push_back(9);  
    cont.push_back(3); cont.push_back(8);  
    cont.push_back(7);  
    // Ausgabe ist 4 9 3 8 7  
    for_each(cont.begin(), cont.end(), Print);  
  
    list<int>::iterator iter =  
        find(cont.begin(), cont.end(), 3);  
    for_each(iter, cont.end(), Print);  
}
```

Ausgabe ?

```
1. 4 9 3 8 7  
2. 3 8 7  
3. 3  
4. 3 4 7 8 9
```

Ergebnis

```
___ 4 9 3 8 7      _ _ 3 8 7  
___ 3      ___ 3 4 7 8 9
```

## Schreiben Sie Code

Implementieren Sie einige STL-Algorithmen selbst, z.B.

- `for_each`
- `find`
- ...
- Bzw. natürlich „leichte“ Abwandlungen davon
  
- Oder in eine Aufgabe, „Implementieren Sie eine Funktion, die ...“
- Implementieren Sie eine Template Klasse `Punkt<dim>`