



Informatik

Übungsstunde 12

Bonus Exercise 2 (stepwise refinement)

1. Input (gegeben)

2. Backtracking

2.1. nächstes Feld

Pseudo Codes

Bonus Exercise 2 (stepwise refinement)

3. Check Solution

Nützliche Funktionen (gegeben)

4. Output (gegeben)

Iteratoren (vector)

```
#include <vector>

std::vector<int> cont = {8,3,1,4,6,9};
for (std::vector<int>::iterator it = cont.begin();
     it != cont.end();
     ++it) {
    std::cout << *it << " ";
}
```

Iteratoren (set)

```
#include <set>

std::set<int> cont = {8,3,1,4,6,9};
for (std::set<int>::iterator it = cont.begin();
     it != cont.end();
     ++it) {
    std::cout << *it << " ";
}
```

Prüfungsaufgabe 1

```
int* j = new int[4]{1,2,0,1};  
int* p = j;  
p += 2;  
std::cout << *(j+1) << std::endl; // Output 1  
std::cout << (p-j) << std::endl; // Output 2  
  
int& k = *(j+2);  
k++;  
(*p)++;  
std::cout << k + *p << std::endl; // Output 3
```

(a) Ausgabe 1/ *Output 1*:

2 P

(b) Ausgabe 2/ *Output 2*:

2 P

(c) Ausgabe 3/ *Output 3*:

2 P

Prüfungsaufgabe 2

```
int* j = new int;
*j = 2;
int& m = *j;
m += 1;
std::cout << m << std::endl; // Output (a)
std::cout << *j << std::endl; // Output (b)

int* p = new int[2]{1,0};
int* o = p;
*(++o) = *j;
std::cout << *p << std::endl; // Output (c)
std::cout << *o << std::endl; // Output (d)

int h = 9;
o = &h;
std::cout << h << std::endl; // Output (e)
std::cout << ++(*o) << std::endl; // Output (f)
```

(a) Ausgabe/*Output*:

1.5 P

(b) Ausgabe/*Output*:

1.5 P

(c) Ausgabe/*Output*:

1.5 P

(d) Ausgabe/*Output*:

1.5 P

(e) Ausgabe/*Output*:

1.5 P

(f) Ausgabe/*Output*:

1.5 P

Prüfungsaufgabe 3

```
1 int* a = new int[6] {1, 3, 5, 7, 9, 11};  
2 int* x = a;  
3 x++;  
4 (*x)++;  
5 std::cout << *(a+*x);
```

Welche Aussage beschreibt die Ausgabe am besten? / Which statement describes the output best?

☐ 8



☐ 7



☐ 11



☐ 9



Prüfungsaufgabe 4

```
1 std::vector<int> v = {1, 2, 3};  
2 for (std::vector<int>::iterator it = v.begin();  
3     it != v.end();  
4     ++it) {  
5  
6     *it = 2 * *it;  
7 }
```

Welche Aussage passt am besten zum Code? / Which statement matches the code best?

- ☐ Kompiliert nicht / Doesn't compile \$
- ☐ Der Code erzeugt einen Speicherzugriffsfehler / The code produces a memory access error \$
- ☐ Nach der Schleife enthält v die Werte 1 2 3 / After the loop, v contains the values 1 2 3 \$
- ☐ Nach der Schleife enthält v die Werte 2 4 6 / After the loop, v contains the values 2 4 6 \$