

Truly Exceptional



```
static int P(std::vector<int>& a, int i){
    int res = a.at(i);
    try{ return res + P(a,i+1); }
    catch (std::out_of_range e) { return 5; }
}
static int Q(std::vector<int>& a, int i){
    try{ return a.at(i) + Q(a,i+1); }
    catch (std::out_of_range e) { return 5; }
}
int main() {
    std::vector<int> a = {1,2,3,4};
    std::cout << P(a,0) << " ";
    std::cout << Q(a,0) << "\n";
    return 0;
}
```

3/27/2017

Was gibt das Programm aus?

- (1)[Fehlermeldung]
- (2)0, dann [Fehlermeldung]
- (3)10, dann [Fehlermeldung]
- (4)11, dann [Fehlermeldung]
- (5)15, dann [Fehlermeldung]
- (6)0, 10
- (7)10,10
- (8)11,11
- (9)11,15
- (10)15,15

Erklärung



```
static int P(std::vector<int>& a, int i){
    int res = a.at(i);
    try{ return res + P(a,i+1); }
    catch (std::out_of_range e) { return 5; }
}
static int Q(std::vector<int>& a, int i){
    try{ return a.at(i) + Q(a,i+1); }
    catch (std::out_of_range e) { return 5; }
}
int main() {
    std::vector<int> a = {1,2,3,4};
    std::cout << P(a,0) << " ";
    std::cout << Q(a,0) << "\n";
    return 0;
}
```

P(a,0) ok bis P(a,3), dann try
return P(a,4). Exception in P(a,4)
[out of bounds!] dadurch catch
block in P(a,3) gibt 5 zurück.

$$5 + 3 + 2 + 1 = 11$$

Erklärung



```
static int P(std::vector<int>& a, int i){
    int res = a.at(i);
    try{ return res + P(a,i+1); }
    catch (std::out_of_range e) { return 5; }
}
static int Q(std::vector<int>& a, int i){
    try{ return a.at(i) + Q(a,i+1); }
    catch (std::out_of_range e) { return 5; }
}
int main() {
    std::vector<int> a = {1,2,3,4};
    std::cout << P(a,0) << " ";
    std::cout << Q(a,0) << "\n";
    return 0;
}
```

Q(a,0) ok bis Q(a,3), dann try
return Q(a,4) in Q(a,4).
Exception in Q(a,4) [out of
bounds!] dadurch **catch block
in Q(a,4)** gibt 5 zurück.

$$5 + 4 + 3 + 2 + 1 = 15$$

Truly Exceptional



```
static int P(std::vector<int>& a, int i){
    int res = a.at(i);
    try{ return res + P(a,i+1); }
    catch (std::out_of_range e) { return 5; }
}
static int Q(std::vector<int>& a, int i){
    try{ return a.at(i) + Q(a,i+1); }
    catch (std::out_of_range e) { return 5; }
}
int main() {
    std::vector<int> a = {1,2,3,4};
    std::cout << P(a,0) << " ";
    std::cout << Q(a,0) << "\n";
    return 0;
}
```

Was gibt das Programm aus?

- (1) [Fehlermeldung]
- (2) 0, dann [Fehlermeldung]
- (3) 10, dann [Fehlermeldung]
- (4) 11, dann [Fehlermeldung]
- (5) 15, dann [Fehlermeldung]
- (6) 0, 10
- (7) 10, 10
- (8) 11, 11
- (9) 11, 15**
- (10) 15, 15