

Code-Puzzle

All Codesnippets originate from the first Round 2016.

Code Snippet 1

Save 2 lines of code and make it more efficient.

```
vector<int> A;
cin >> N;
for(int i = 0; i < N; ++i) {
    int temp;
    cin >> temp;
    A.push_back(temp);
}
```

Code Snippet 2

What could possibly go wrong?

```
while(*minCeilings.begin() - *maxFloors.begin() <= 0
      && !minCeilings.empty()) {
    maxFloors.erase(maxFloors.find(floors[tail]));
    minCeilings.erase(minCeilings.find(ceilings[tail]));
    tail ++;
    currentSight --;
}
```

Code Snippet 3

Make it shorter and more readable.

```
bool even = true;
for(int j = 0; j<amount; j++) {
    cin >> vec[j];
    if(j!=0) {
        dist=vec[j]-vec[j-1];
    }
    if(j>1) {
        if(dist!=vec[j-1]-vec[j-2]) {
            even = false;
            break;
        }
    }
}
```

Code Snippet 4

Make it shorter and clearer.

```
if(diff!=(vec[j+1]-last))
{
    even = false;
}
else
{
    even = true;
}
if(even)
{
    last = vec[j+1];
    len++;
}
if(len > longest_len)
{
    longest_len = len;
}
```

Code Snippet 5

Write this function in a more elegant way.

```
for (int j = i; j <= N; ++j){
    if(d[j] < mnD){
        mnD = d[j];
    }
    if(b[j] > mxB){
        mxB = b[j];
    }
    if (mxB >= mnD){
        temp = j;
        break;
    }
}
```

Code Snippet 6

Rewrite both loops as for-loops.

```
int i = 0;
int mx = 1;
while(i < N - 1){
    int m = v[i] - v[i + 1];
    ct = 2;
    ++i;
    while(i < N - 1){
        if(v[i] - v[i + 1] == m){
            ++ct;
        } else{
            break;
        }
        i++;
    }
    if (ct > mx){
        mx = ct;
    }
}
```

Code Snippet 7

Use the STL luke!

```
vector < vector < int > > vla;

int minimum(int c)
{
    int minim = 1000000;
    for(int y = 0; y<num; y++)
        if (c != y)
            minim = min(minim, vla[c][y]);
    return minim;
}

int main() {
    ...
    minimum(v1)
    ...
}
```

Code Snippet 8

Use a range based for-loop und shorten the code to 7 lines.

```
set<int>::iterator it = times.begin();
int start=-1;
int count = 0;
for(unsigned int j=1;j<times.size();j++){
    for(unsigned int k=0;k < schwings.size();k++){
        if(schwings[k].isPlaying(*it)){
            count++;
            if(start==-1){
                start=*it;
            }
            break;
        }
    }
    it++;
}
cout << count << " " << start << endl;
```

Code Snippet 9

With another approach this code takes 7 lines of code.

```
cin >> n;
for(int j = 0; j < n; j++){
    cin >> temp;
    w.push_back(temp);
}
for(int j = 0; j < n; j++){
    cin >> temp;
    h.push_back(temp);
}
s = false;
while(s==false){
    for(int j = 1; j < n; j++){
        if(w.at(j-1) < w.at(j)){
            temp = w.at(j-1);
            w.at(j-1)=w.at(j);
            w.at(j)=temp;
            temp = h.at(j-1);
            h.at(j-1)=h.at(j);
            h.at(j)=temp;
        }
    }
    s = true;
    for(int j = 1; j < n; j++){
        if(w.at(j-1) < w.at(j)){
            s = false;
            break;
        }
    }
}
```