

1) Das Josephus-Problem

```
#include <iostream>

struct Person {
    int    i;
    Person* p;
};

Person* build(int n) {
    Person *prev = new Person(), *first = prev;
    first->i = 1;
    for(int i = 2; i <= n; ++i) {
        prev = prev->p = new Person();
        prev->i = i;
    }
    // close ring and return pointer to first element
    return prev->p = first;
}

void josephus(Person *&ring, int m) {
    // run to one before the guy with bad luck
    for(int i = 2; i < m; ++i)
        ring = ring->p;

    // save the element for deletion
    Person* del = ring->p;
    // update link and go to next
    ring = ring->p = del->p;

    std::cout << del->i << " ";
    delete del;

    if(ring->p != ring)
        josephus(ring, m);
}

int main() {
    int n, m;

    std::cout << "n = "; std::cin >> n;
    std::cout << "m = "; std::cin >> m;

    Person* ring = build(n);
    josephus(ring, m);
    std::cout << "josephus:" << ring->i << std::endl;

    // josephus must die
    delete ring;
}
```