

```
1 #include <iostream>
2 #include <cmath>
3 #include <vector>
4 #include <algorithm>
5
6 // By this point in the course standard classes weren't introduced.
7 // Nevertheless it is simpler to manage an array of arbitrary size by using
8 // the class std::vector, which manages the memory allocation internally.
9
10 // It also makes it easier to get the maximum element.
11
12 int cont(std::vector<unsigned> v, int m, int M, int n);
13 void compute_intervals(const std::vector<unsigned>, unsigned, unsigned, int);
14
15 int main() {
16     unsigned l, n;
17
18     std::cout << "Number of intervals: ";
19     std::cin >> l;
20
21     std::cout << "Size of dataset: ";
22     std::cin >> n;
23
24     std::cout << "Input data: ";
25
26     std::vector<unsigned> v;
27     unsigned x;
28     for (auto t = 0; t < n; t++) {
29         std::cin >> x;
30         v.push_back(x);
31     }
32
33     int MAX = *max_element(std::begin(v), std::end(v));
34
35     // COMPUTE INTERVALS
36     compute_intervals(v, n, l, MAX);
37
38     return 0;
39 }
40
41
42 int cont(std::vector<unsigned> v, int m, int M, int n) {
43     int tot = 0;
44     for (auto n : v)
45         if (n >= m && n < M)
46             tot++;
47     return tot;
48 }
49
50
51 void compute_intervals(const std::vector<unsigned> v, unsigned n, unsigned l, int M) {
52     int k, i;
53     k = ceil((double)M / l);
54     for (i = 0; i < (l - 1); i++)
55         std::cout << i * k << " : " << cont(v, (i * k), ((i + 1) * k), n) << "\n";
56     std::cout << i * k << " : " << cont(v, (i * k), (M + 1), n) << "\n";
57 }
```