```
#include <cmath>
#include <iostream>

#include "polynomial.h"
#include "polynomialring.h"

void addition_table(int modulus, int max_order, int split_at) {
        PolynomialRing r = *generateElements(modulus, max_order);
        int elements = r.size();

        int tables = (int)ceil(1.0*elements / split_at);
        int min = 0;
        for (int split_n = 0; split_n < tables; split_n++, min += split_at) {
                int max = min + split_at;

                /* Start table */
                std::cout << "\\begin{tabular}{p{2.5cm}|";
                for (int i = 0; i < split_at; i++) std::cout << "|p{2.5cm}";
                std::cout << "}\n";

                std::cout << "$+$";
                int c = 0;
                for (PolynomialRing::iterator i = r.begin(); c < elements; c++,
i++) {
                        if (c < min || c >= max) continue;
                        std::cout << "&$" << *i << "$";
                }
                std::cout << "\\\\\\hline\\hline\n";

                for (PolynomialRing::iterator i = r.begin(); i != r.end(); i++)
{
                        std::cout << "$" << *i << "$";
                        c = 0;
                        for (PolynomialRing::iterator j = r.begin(); c < element
s; c++, j++) {
                                if (c < min || c >= max) continue;
                                std::cout << "&$" << *i + *j << "$";
                        }
                        std::cout << "\\\\\n";
                }

                /* End table */
                std::cout << "\\end{tabular}\n\n";
        }

}

void multiplication_table(int modulus, int max_order, int split_at, Polynomial&
mod) {
        PolynomialRing r = *generateElements(modulus, max_order);
        int elements = r.size();

        int tables = (int)ceil(1.0*elements / split_at);
        int min = 0;
        for (int split_n = 0; split_n < tables; split_n++, min += split_at) {
                int max = min + split_at;
                int c;

                /* Start table */
                std::cout << "\\begin{tabular}{p{2.5cm}|";
                c = 0;
                for (int i = 0; i < elements; i++, c++) {
```



```
                        if (c < min || c >= max) continue;
                        std::cout << "|p{2.5cm}";
                }
                std::cout << "}\n";

                std::cout << "$\\times$";
                c = 0;
                for (PolynomialRing::iterator i = r.begin(); c < elements; c++,
i++) {
                        if (c < min || c >= max) continue;
                        std::cout << "&$" << *i << "$";
                }
                std::cout << "\\\\\\hline\\hline\n";

                for (PolynomialRing::iterator i = r.begin(); i != r.end(); i++)
{
                        std::cout << "$" << *i << "$";
                        c = 0;
                        for (PolynomialRing::iterator j = r.begin(); c < element
s; c++, j++) {
                                if (c < min || c >= max) continue;
                                std::cout << "&$" << (*i * *j)%mod << "$";
                        }
                        std::cout << "\\\\\n";
                }

                /* End table */
                std::cout << "\\end{tabular}\n\n";
        }
}
```