```c
#include <stdio.h>
#include <stdlib.h>

// Function to print an element in tuple form
void print_tuple(int num, int q, int p) {
    int tuple[p];
    for (int i = 0; i < p; i++) {
        tuple[i] = num % q;
        num /= q;
    }
    printf("(");
    for (int i = p - 1; i >= 0; i--) {
        printf("%d", tuple[i]);
        if (i > 0) printf(", ");
    }
    printf(")");
}

// Function to print polynomial representation
void print_polynomial(int num, int p) {
    int first = 1;
    for (int i = p - 1; i >= 0; i--) {
        if ((num / (1 << i)) % 2) { // Extract coefficient
            if (!first) printf(" + ");
            if (i == 0) printf("1");
            else if (i == 1) printf("x");
            else printf("x^%d", i);
            first = 0;
        }
    }
    if (first) printf("0");
```

```
}

int main() {
    int q, p;
    printf("Enter base field size q: ");
    scanf("%d", &q);
    printf("Enter extension degree p: ");
    scanf("%d", &p);

    if (q < 2 || p < 1) {
        printf("Invalid input. q must be >= 2 and p >= 1.\n");
        return 1;
    }

    int total_elements = 1;
    for (int i = 0; i < p; i++) {
        total_elements *= q;
    }

    printf("\nElements of GF(%d^%d):\n", q, p);
    printf("------------------------------------------------------\n");
    printf("| Decimal | Tuple Form | Polynomial Form                  |\n");
    printf("------------------------------------------------------\n");
    for (int num = 0; num < total_elements; num++) {
        printf("| %7d | ", num);
        print_tuple(num, q, p);
        printf(" | ");
        print_polynomial(num, p);
        printf(" |\n");
    }
    printf("------------------------------------------------------\n");
    return 0;
}
```