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

int power(int base, int exp, int mod) {
    int result = 1;
    base = base % mod;
    while (exp > 0) {
        if (exp % 2 == 1) {
            result = (result * base) % mod;
        }
        exp = exp >> 1;
        base = (base * base) % mod;
    }
    return result;
}

int is_primitive_root(int candidate, int p) {
    for (int i = 1; i < p - 1; i++) {
        if (power(candidate, i, p) == 1) {
            return 0;
        }
    }
    return 1;
}

int find_primitive_root(int p) {
    for (int g = 2; g < p; g++) {
        if (is_primitive_root(g, p)) {
            return g;
        }
    }
    return -1;
}

int main() {
    int p;
    printf("Enter a prime number: ");
    scanf("%d", &p);

    if (p <= 1) {
        printf("Input must be a prime number greater than 1\n");
        return 0;
    }

    int root = find_primitive_root(p);
    if (root == -1) {
        printf("No primitive root found for the prime number %d\n", p);
    } else {
        printf("A primitive root of %d is: %d\n", p, root);
    }

    return 0;
}
```