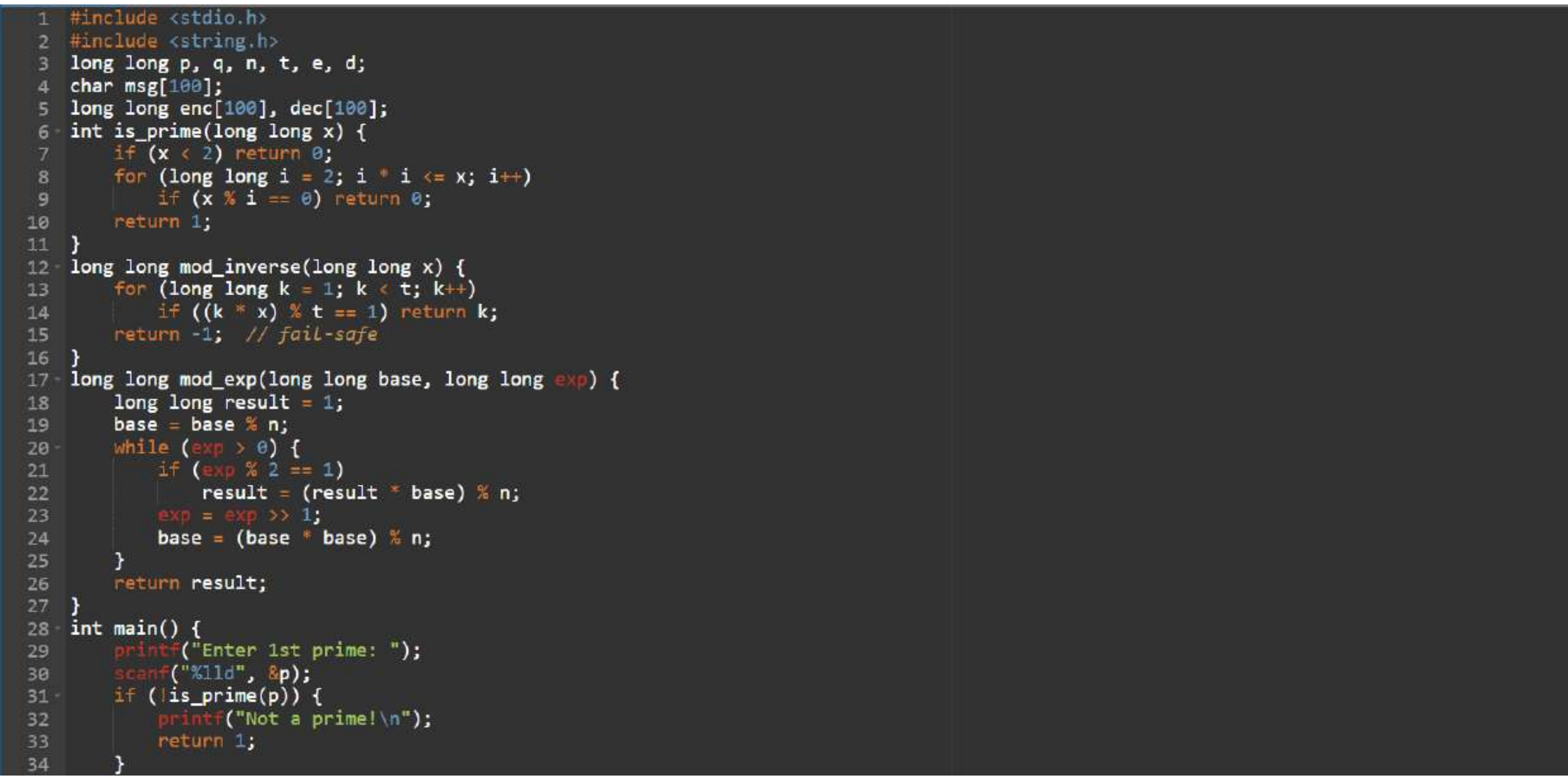
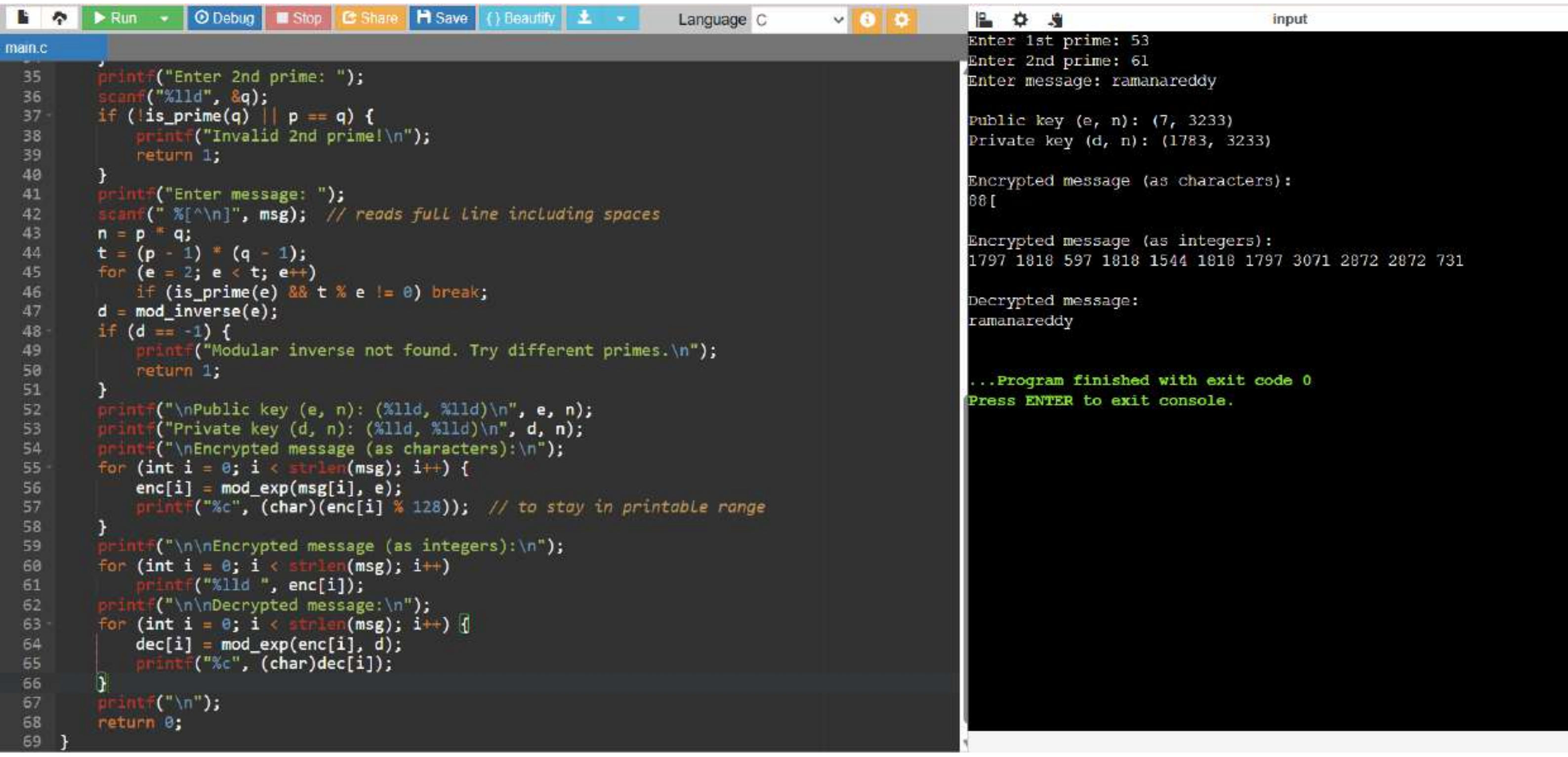


```c
#include <stdio.h>
#include <string.h>
long long p, q, n, t, e, d;
char msg[100];
long long enc[100], dec[100];
int is_prime(long long x) {
    if (x < 2) return 0;
    for (long long i = 2; i * i <= x; i++)
        if (x % i == 0) return 0;
    return 1;
}
long long mod_inverse(long long x) {
    for (long long k = 1; k < t; k++)
        if ((k * x) % t == 1) return k;
    return -1;  // fail-safe
}
long long mod_exp(long long base, long long exp) {
    long long result = 1;
    base = base % n;
    while (exp > 0) {
        if (exp % 2 == 1)
            result = (result * base) % n;
        exp = exp >> 1;
        base = (base * base) % n;
    }
    return result;
}
int main() {
    printf("Enter 1st prime: ");
    scanf("%lld", &p);
    if (!is_prime(p)) {
        printf("Not a prime!\n");
        return 1;
    }
```



```c
    }
    printf("Enter 2nd prime: ");
    scanf("%lld", &q);
    if (!is_prime(q) || p == q) {
        printf("Invalid 2nd prime!\n");
        return 1;
    }
    printf("Enter message: ");
    scanf(" %[^\n]", msg);  // reads full line including spaces
    n = p * q;
    t = (p - 1) * (q - 1);
    for (e = 2; e < t; e++)
        if (is_prime(e) && t % e != 0) break;
    d = mod_inverse(e);
    if (d == -1) {
        printf("Modular inverse not found. Try different primes.\n");
        return 1;
    }
    printf("\nPublic key (e, n): (%lld, %lld)\n", e, n);
    printf("Private key (d, n): (%lld, %lld)\n", d, n);
    printf("\nEncrypted message (as characters):\n");
    for (int i = 0; i < strlen(msg); i++) {
        enc[i] = mod_exp(msg[i], e);
        printf("%c", (char)(enc[i] % 128));  // to stay in printable range
    }
    printf("\n\nEncrypted message (as integers):\n");
    for (int i = 0; i < strlen(msg); i++)
        printf("%lld ", enc[i]);
    printf("\n\nDecrypted message:\n");
    for (int i = 0; i < strlen(msg); i++) {
        dec[i] = mod_exp(enc[i], d);
        printf("%c", (char)dec[i]);
    }
    printf("\n");
    return 0;
}
```

```
Enter 1st prime: 53
Enter 2nd prime: 61
Enter message: ramanareddy

Public key (e, n): (7, 3233)
Private key (d, n): (1783, 3233)

Encrypted message (as characters):
88[

Encrypted message (as integers):
1797 1818 597 1818 1544 1818 1797 3071 2872 2872 731

Decrypted message:
ramanareddy


...Program finished with exit code 0
Press ENTER to exit console.
```