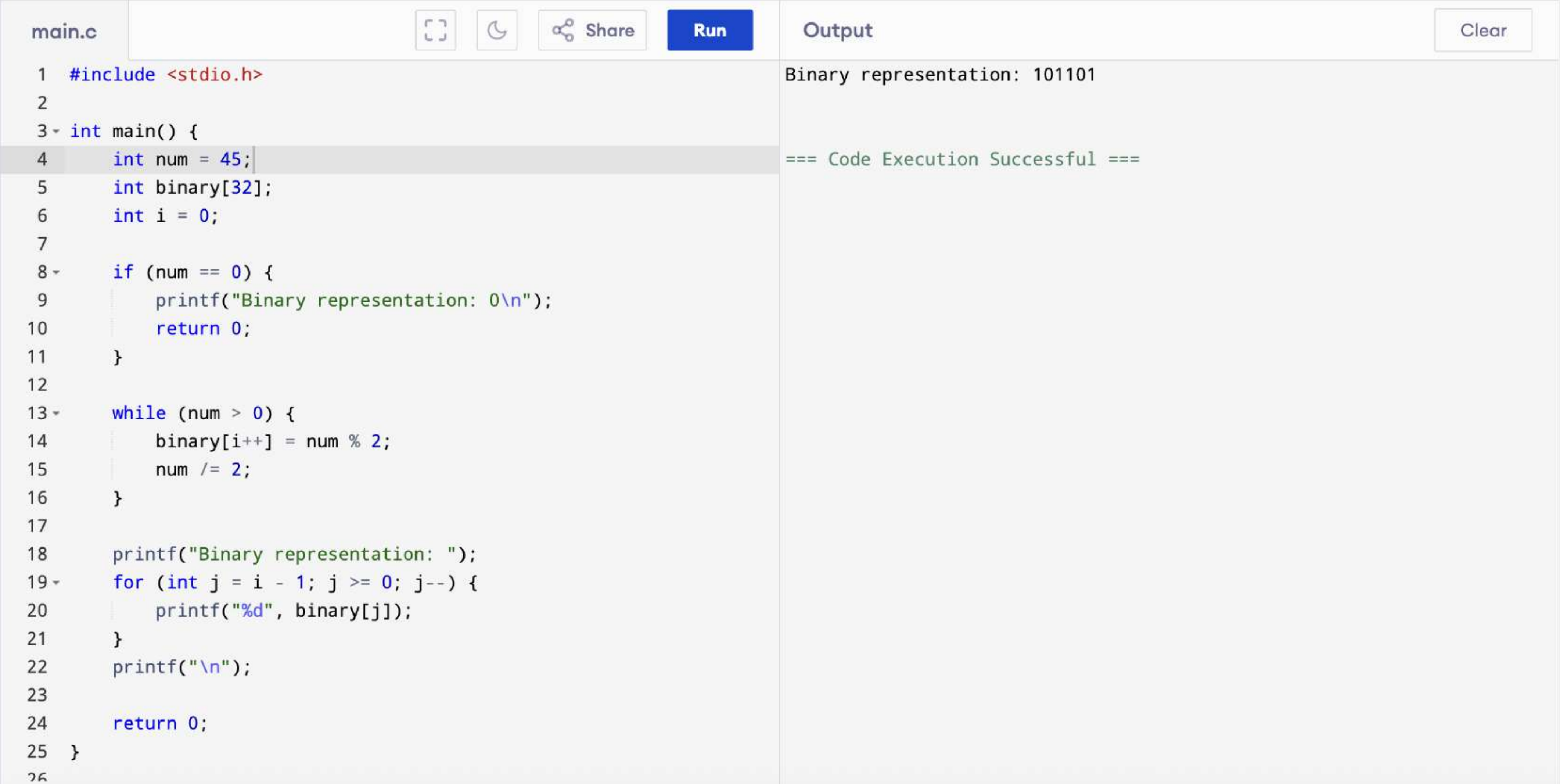
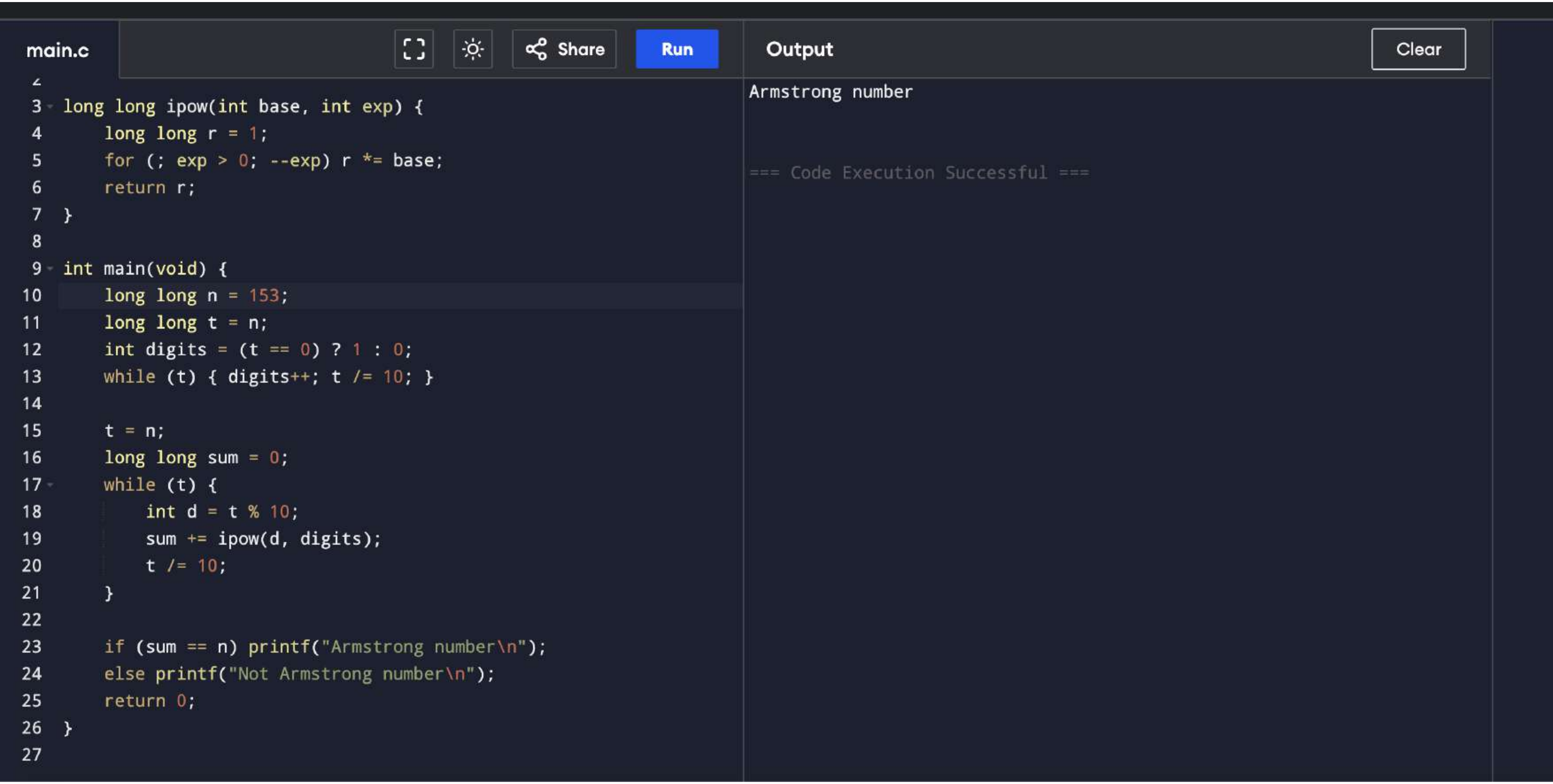


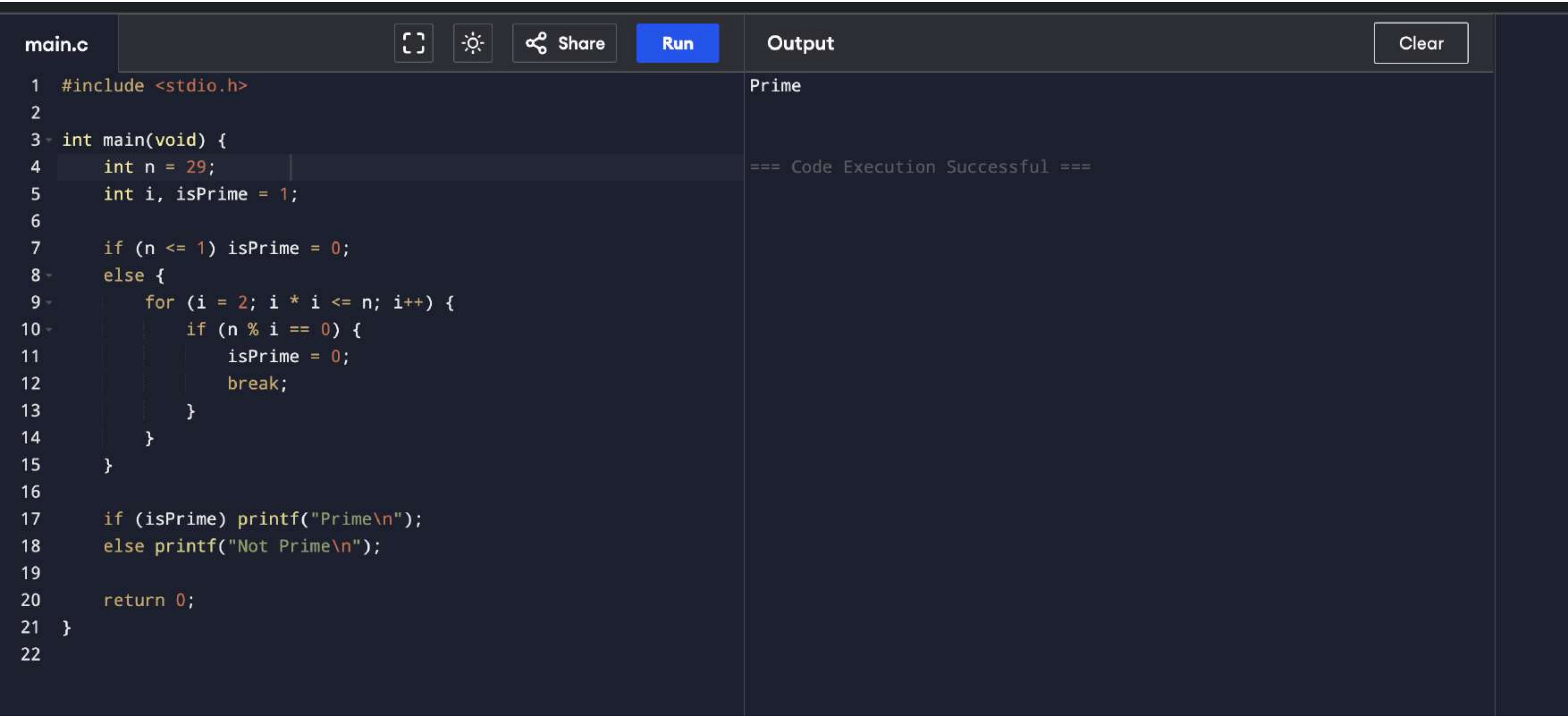
```c
#include <stdio.h>

int main() {
    int num = 45;
    int binary[32];
    int i = 0;

    if (num == 0) {
        printf("Binary representation: 0\n");
        return 0;
    }

    while (num > 0) {
        binary[i++] = num % 2;
        num /= 2;
    }

    printf("Binary representation: ");
    for (int j = i - 1; j >= 0; j--) {
        printf("%d", binary[j]);
    }
    printf("\n");

    return 0;
}
```

```
Binary representation: 101101


=== Code Execution Successful ===
```



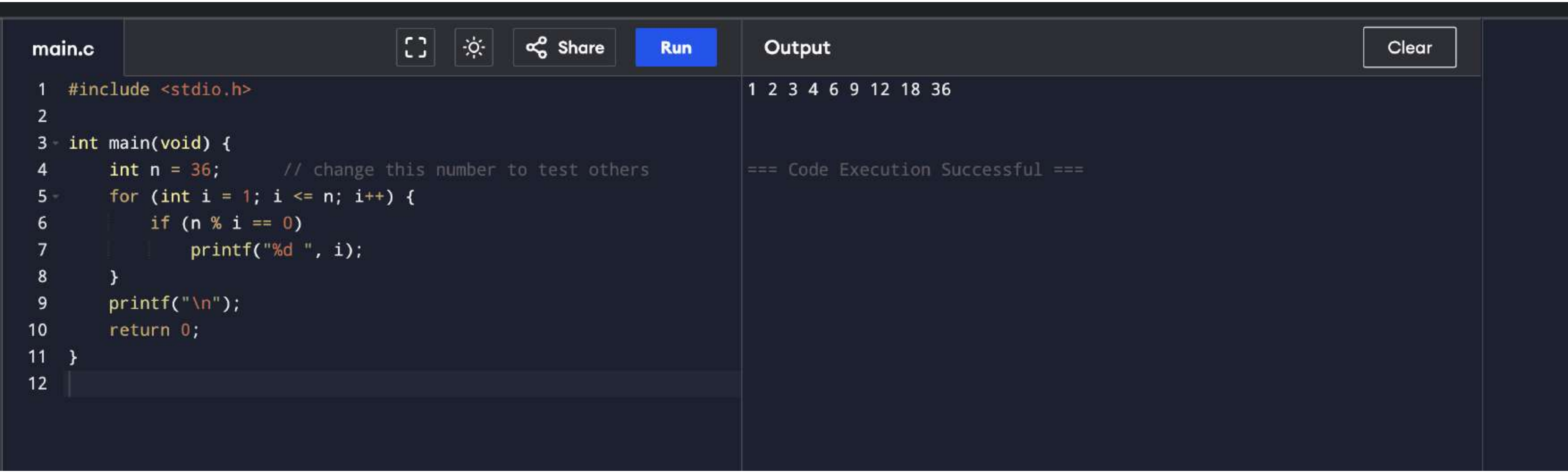
```c
long long ipow(int base, int exp) {
    long long r = 1;
    for (; exp > 0; --exp) r *= base;
    return r;
}

int main(void) {
    long long n = 153;
    long long t = n;
    int digits = (t == 0) ? 1 : 0;
    while (t) { digits++; t /= 10; }

    t = n;
    long long sum = 0;
    while (t) {
        int d = t % 10;
        sum += ipow(d, digits);
        t /= 10;
    }

    if (sum == n) printf("Armstrong number\n");
    else printf("Not Armstrong number\n");
    return 0;
}

```

```
Armstrong number


=== Code Execution Successful ===
```



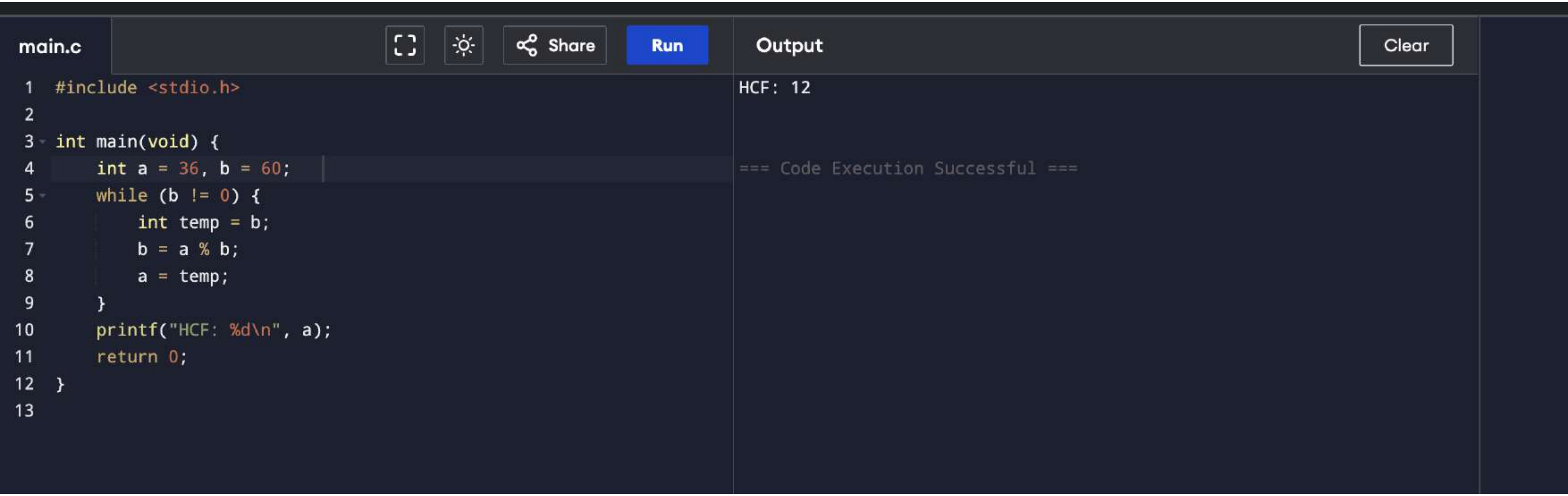
```c
#include <stdio.h>

int main(void) {
    int n = 29;
    int i, isPrime = 1;

    if (n <= 1) isPrime = 0;
    else {
        for (i = 2; i * i <= n; i++) {
            if (n % i == 0) {
                isPrime = 0;
                break;
            }
        }
    }

    if (isPrime) printf("Prime\n");
    else printf("Not Prime\n");

    return 0;
}

```

```
Prime


=== Code Execution Successful ===
```



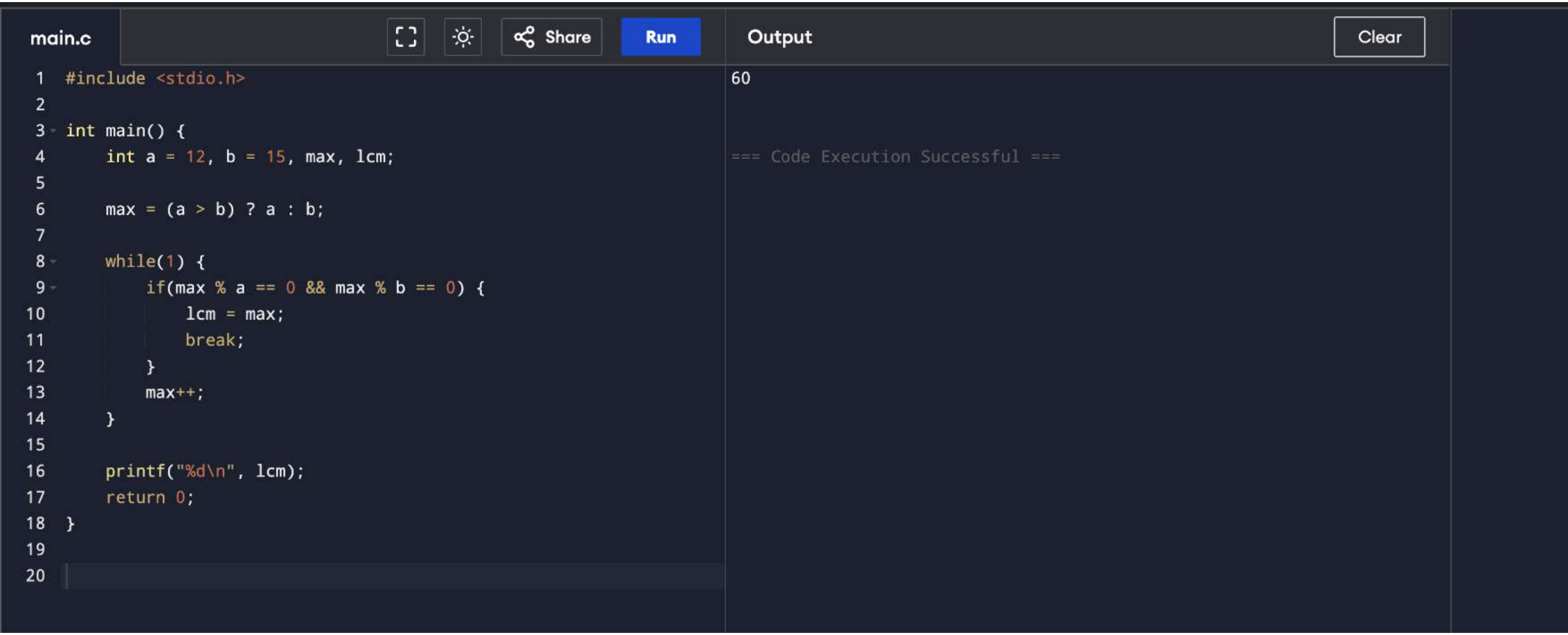
```c
#include <stdio.h>

int main(void) {
    int n = 36;      // change this number to test others
    for (int i = 1; i <= n; i++) {
        if (n % i == 0)
            printf("%d ", i);
    }
    printf("\n");
    return 0;
}

```

```
1 2 3 4 6 9 12 18 36


=== Code Execution Successful ===
```



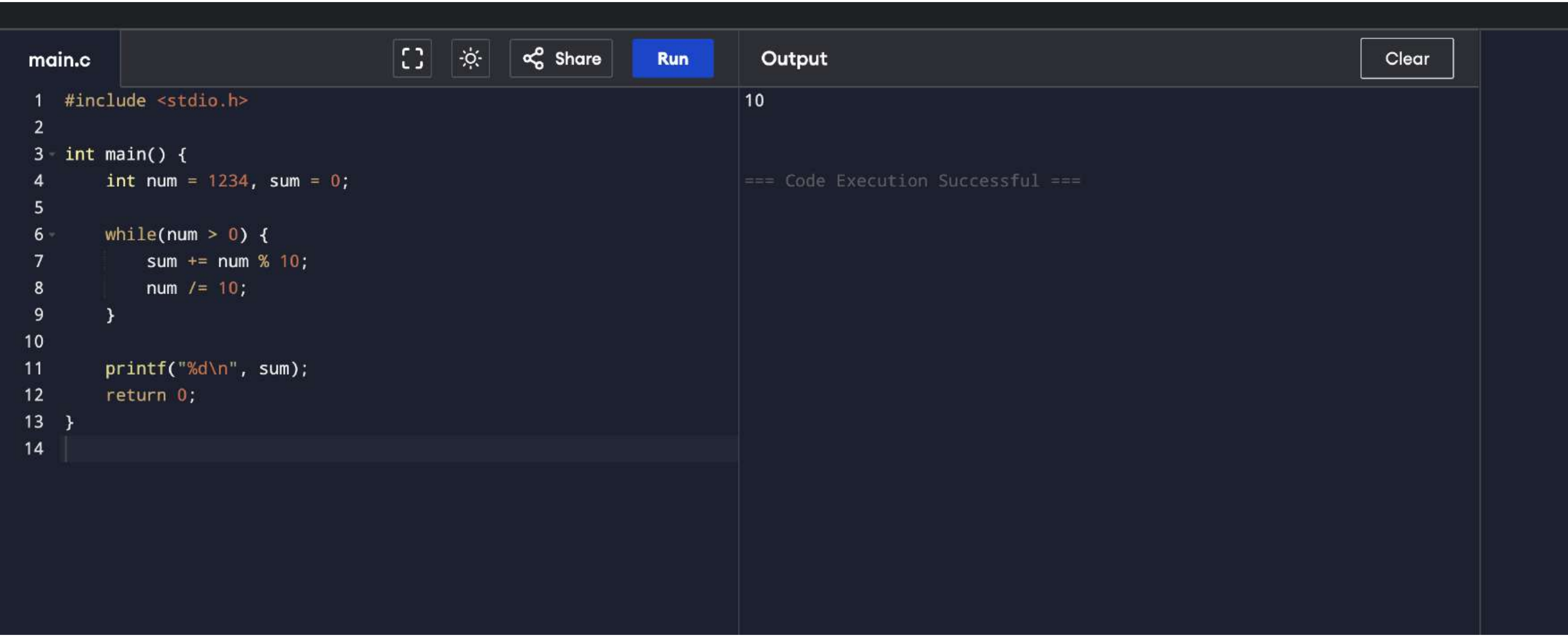
```c
#include <stdio.h>

int main(void) {
    int a = 36, b = 60;
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    printf("HCF: %d\n", a);
    return 0;
}
```

```
HCF: 12


=== Code Execution Successful ===
```



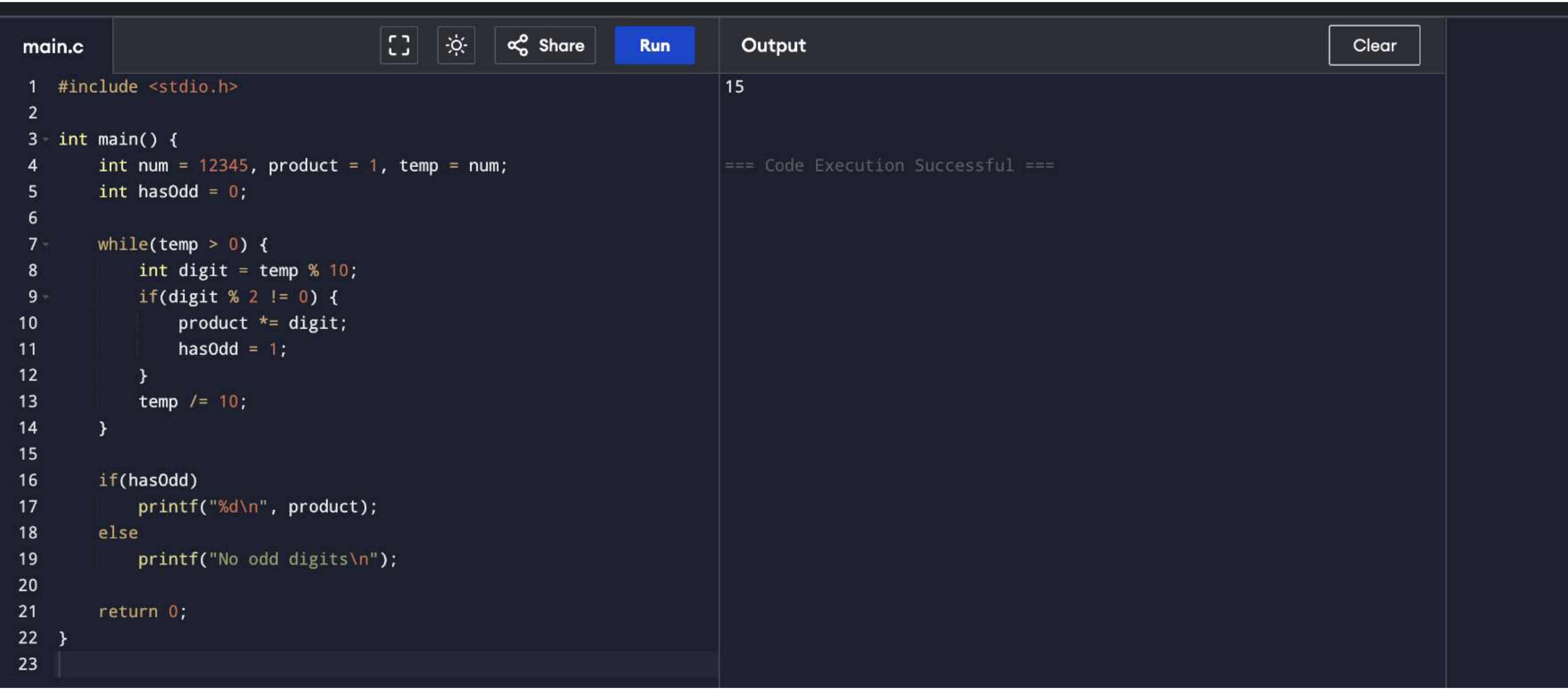
```c
#include <stdio.h>

int main() {
    int a = 12, b = 15, max, lcm;

    max = (a > b) ? a : b;

    while(1) {
        if(max % a == 0 && max % b == 0) {
            lcm = max;
            break;
        }
        max++;
    }

    printf("%d\n", lcm);
    return 0;
}

```

```
60


=== Code Execution Successful ===
```



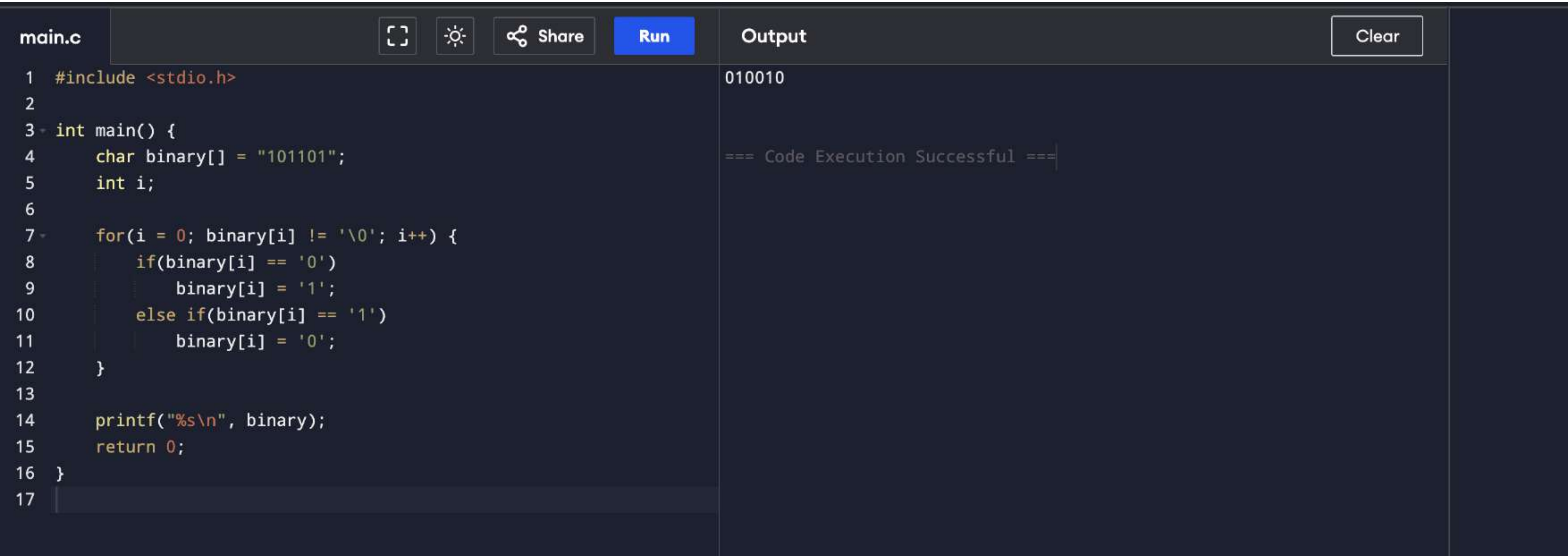
```c
#include <stdio.h>

int main() {
    int num = 1234, sum = 0;

    while(num > 0) {
        sum += num % 10;
        num /= 10;
    }

    printf("%d\n", sum);
    return 0;
}

```

```
10


=== Code Execution Successful ===
```



```c
#include <stdio.h>

int main() {
    int num = 12345, product = 1, temp = num;
    int hasOdd = 0;

    while(temp > 0) {
        int digit = temp % 10;
        if(digit % 2 != 0) {
            product *= digit;
            hasOdd = 1;
        }
        temp /= 10;
    }

    if(hasOdd)
        printf("%d\n", product);
    else
        printf("No odd digits\n");

    return 0;
}
```

```
15


=== Code Execution Successful ===
```



```c
#include <stdio.h>

int main() {
    char binary[] = "101101";
    int i;

    for(i = 0; binary[i] != '\0'; i++) {
        if(binary[i] == '0')
            binary[i] = '1';
        else if(binary[i] == '1')
            binary[i] = '0';
    }

    printf("%s\n", binary);
    return 0;
}
```

```
010010


=== Code Execution Successful ===
```