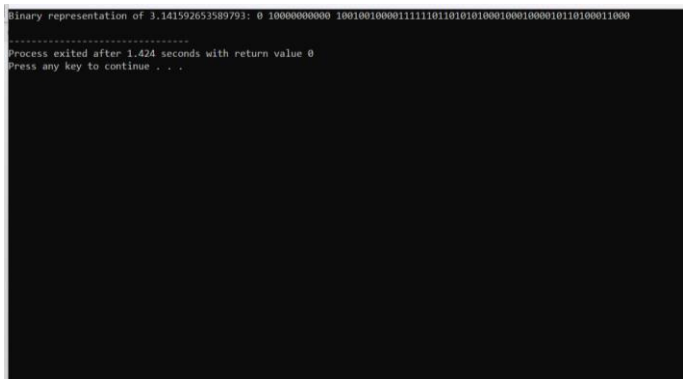
Double-precision representation:

```
#include <stdio.h>
#include <stdint.h>
void print_double_binary(double num) {
    uint64_t *ptr = (uint64_t *)&num; // Treat the double as a 64-bit unsigned integer
    uint64_t mask = 1ULL << 63; // Start with the most significant bit

    printf("Binary representation of %.15lf: ", num);

    for (int i = 0; i < 64; i++) {
        printf("%d", (*ptr & mask) ? 1 : 0);
        if (i == 0 || i == 11) // Print the sign bit and the exponent
            printf(" ");
        mask >>= 1; // Move to the next bit
    }
    printf("\n");
}

int main() {
    double num = 3.141592653589793238; // Example double-precision floating-point number

    print_double_binary(num); // Print the binary representation

    return 0;
}
```

Input&output:


Binary representation of 3.141592653589793: 0 10000000000 1001001000011111101101010001000100001011010001 1000

Process exited after 1.424 seconds with return value 0
Press any key to continue . . .