#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <stdint.h>

#include <math.h> // Required for sin() and fabs()

// Constants for MD5

uint32\_t A, B, C, D;

// MD5 auxiliary functions

#define F(x, y, z) ((x & y) | (~x & z))

#define G(x, y, z) ((x & z) | (y & ~z))

#define H(x, y, z) (x ^ y ^ z)

#define I(x, y, z) (y ^ (x | ~z))

#define LEFTROTATE(x, c) (((x) << (c)) | ((x) >> (32 - (c))))

// Sine-based constants

uint32\_t K[64];

int s[] = {

7,12,17,22, 7,12,17,22, 7,12,17,22, 7,12,17,22,

5, 9,14,20, 5, 9,14,20, 5, 9,14,20, 5, 9,14,20,

4,11,16,23, 4,11,16,23, 4,11,16,23, 4,11,16,23,

6,10,15,21, 6,10,15,21, 6,10,15,21, 6,10,15,21

};

// Initialize K with sine-derived constants

void init\_K() {

for (int i = 0; i < 64; i++) {

K[i] = (uint32\_t)(fabs(sin(i + 1)) \* 4294967296.0);

}

}

// Process one 512-bit block

void md5\_block(uint32\_t \*M) {

uint32\_t a = A, b = B, c = C, d = D, f, g, temp;

for (int i = 0; i < 64; i++) {

if (i < 16) {

f = F(b, c, d);

g = i;

} else if (i < 32) {

f = G(b, c, d);

g = (5 \* i + 1) % 16;

} else if (i < 48) {

f = H(b, c, d);

g = (3 \* i + 5) % 16;

} else {

f = I(b, c, d);

g = (7 \* i) % 16;

}

temp = d;

d = c;

c = b;

b = b + LEFTROTATE((a + f + K[i] + M[g]), s[i]);

a = temp;

}

A += a;

B += b;

C += c;

D += d;

}

// Pads the message and runs MD5

void md5(unsigned char \*initial\_msg, size\_t initial\_len) {

uint8\_t \*msg = NULL;

// Calculate new length with padding

size\_t new\_len = ((((initial\_len + 8) / 64) + 1) \* 64);

msg = (uint8\_t \*)calloc(new\_len, 1); // Zero-padded

memcpy(msg, initial\_msg, initial\_len);

msg[initial\_len] = 0x80; // append the '1' bit

// Append message length in bits (little endian)

uint64\_t bits\_len = 8 \* initial\_len;

memcpy(msg + new\_len - 8, &bits\_len, 8);

// Initialize buffers

A = 0x67452301;

B = 0xefcdab89;

C = 0x98badcfe;

D = 0x10325476;

// Process each 512-bit chunk

for (size\_t offset = 0; offset < new\_len; offset += 64) {

md5\_block((uint32\_t \*)(msg + offset));

}

// Output the final digest

printf("%08x%08x%08x%08x\n", A, B, C, D);

free(msg);

}

int main() {

init\_K();

char msg[] = "hello world";

printf("MD5(\"%s\") = ", msg);

md5((unsigned char\*)msg, strlen(msg));

return 0;

}

A screenshot of a computer

AI-generated content may be incorrect.