9) To write a python program to implement the MD5 hashing technique.

PROGRAM:-

```
import math, struct
def L(x, n): return ((x << n) | (x >> (32 - n))) & 0xFFFFFFFF
def F0(abcd): return (abcd[1] & abcd[2]) | (~abcd[1] & abcd[3])
def F1(abcd): return (abcd[3] & abcd[1]) | (~abcd[3] & abcd[2])
def F2(abcd): return abcd[1] ^ abcd[2] ^ abcd[3]
def F3(abcd): return abcd[2] ^ (abcd[1] | ~abcd[3])
def md5(msg: bytes) -> str:
    h = [0x67452301, 0xEFCDAB89, 0x98BADCFE, 0x10325476]
    orig_len_bits = len(msg) * 8
    msg += b'\x80' + b'\x00' * ((56 - (len(msg) + 1) % 64) % 64)
    msg += struct.pack('<Q', orig_len_bits)
    k = [int(abs(math.sin(i + 1)) * 2**32) & 0xFFFFFFFF for i in range(64)]
    s = [[7,12,17,22],[5,9,14,20],[4,11,16,23],[6,10,15,21]]
    fns, M, O = [F0, F1, F2, F3], [1,5,3,7], [0,1,5,0]
    for i in range(0, len(msg), 64):
        a,b,c,d = h
        w = list(struct.unpack('<16I', msg[i:i+64]))
        for p in range(4):
            for q in range(16):
                g = (M[p]*q + O[p]) % 16
                f = fns[p]([a,b,c,d])
                tmp = (a + f + k[q+16*p] + w[g]) & 0xFFFFFFFF
                tmp = L(tmp, s[p][q % 4])
                a,b,c,d = d, (b + tmp) & 0xFFFFFFFF, b, c
        h = [(x + y) & 0xFFFFFFFF for x, y in zip(h, [a,b,c,d])]
    return ''.join(f'{v:02x}' for x in h for v in struct.pack('<I', x))
msg = b"The quick brown fox jumps over the lazy dog"
print("0x" + md5(msg))
```

OUTPUT:-

Output:

```
0x9e107d9d372bb6826bd81d3542a419d6
```
