IMPLEMENTATION OF MD5

Cryptographic Hash Function

AIM

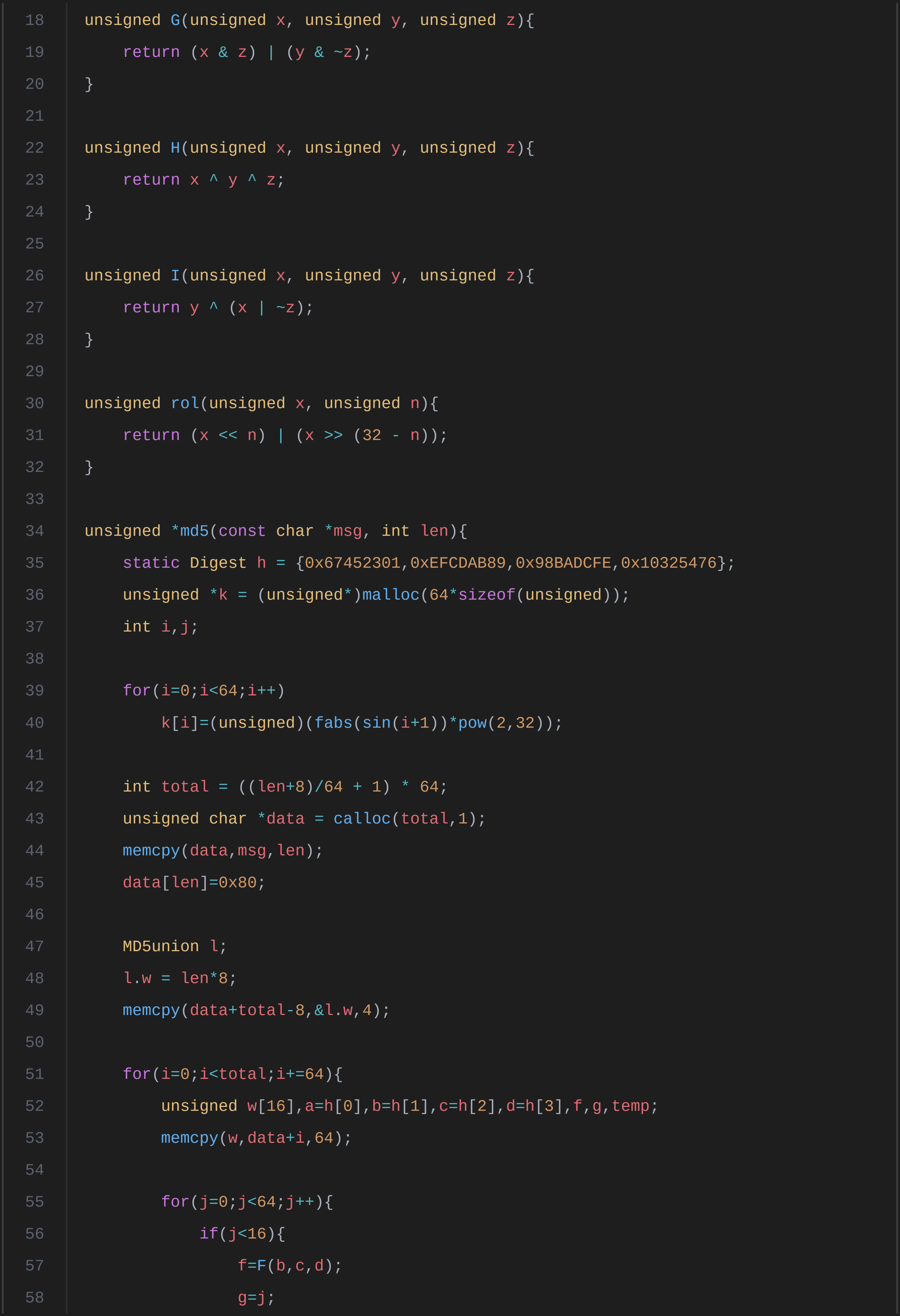
To write a C program to implement the MD5 hashing technique.

ALGORITHM

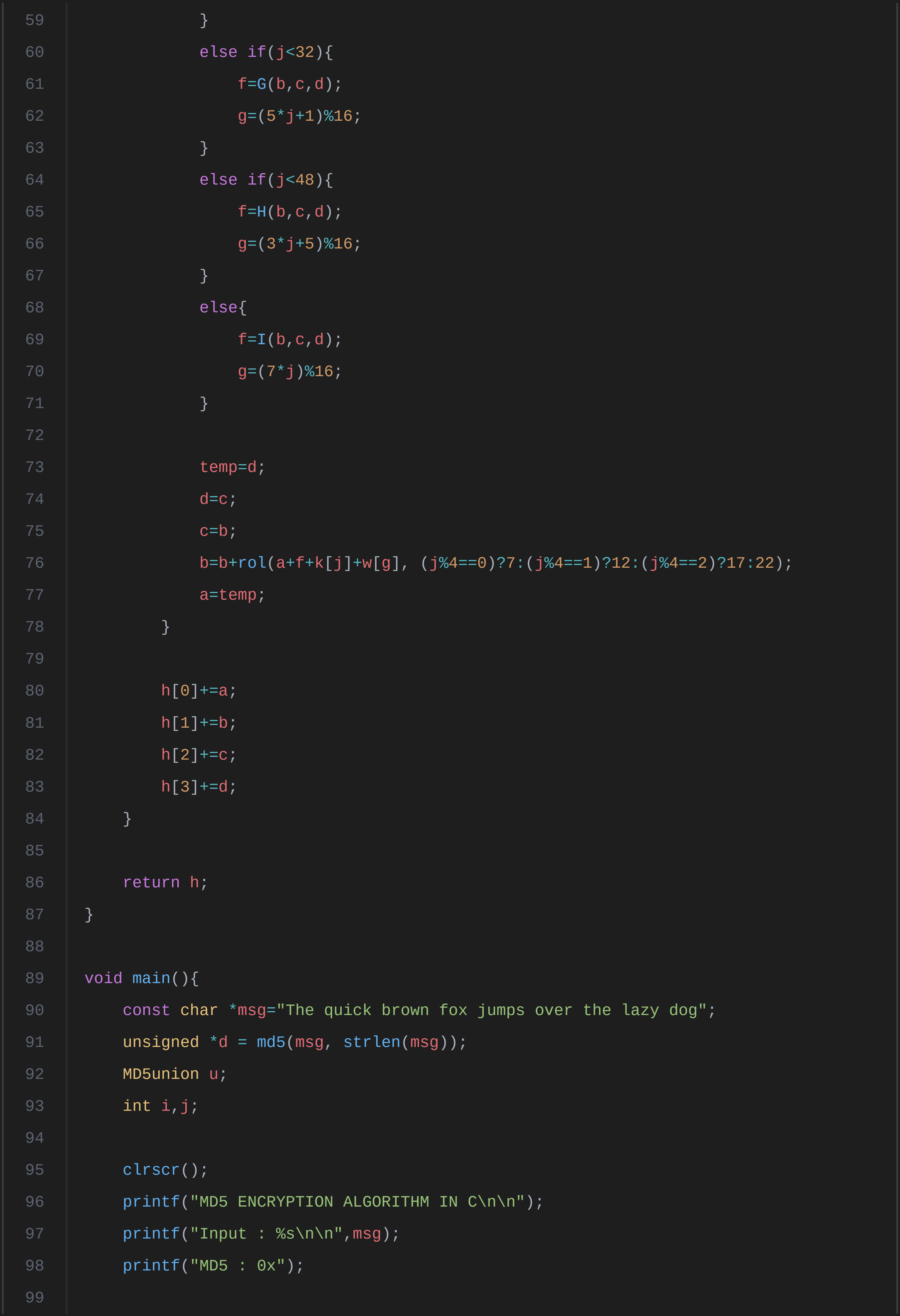
- **STEP 1:** Take the input message.
- **STEP 2:** Pad the message so its length is a multiple of 512 bits.
- **STEP 3:** Initialize four 32-bit registers A, B, C, D.
- **STEP 4:** Process each 512-bit block using functions F, G, H, I and circular shifts.
- **STEP 5:** Combine the results to produce a 128-bit hash output.

PROGRAM

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <conio.h>

typedef union {
    unsigned w;
    unsigned char b[4];
} MD5union;

typedef unsigned Digest[4];

unsigned F(unsigned x, unsigned y, unsigned z){
    return (x & y) | (~x & z);
}

```



```c
unsigned G(unsigned x, unsigned y, unsigned z){
    return (x & z) | (y & ~z);
}

unsigned H(unsigned x, unsigned y, unsigned z){
    return x ^ y ^ z;
}

unsigned I(unsigned x, unsigned y, unsigned z){
    return y ^ (x | ~z);
}

unsigned rol(unsigned x, unsigned n){
    return (x << n) | (x >> (32 - n));
}

unsigned *md5(const char *msg, int len){
    static Digest h = {0x67452301,0xEFCDAB89,0x98BADCFE,0x10325476};
    unsigned *k = (unsigned*)malloc(64*sizeof(unsigned));
    int i,j;

    for(i=0;i<64;i++)
        k[i]=(unsigned)(fabs(sin(i+1))*pow(2,32));

    int total = ((len+8)/64 + 1) * 64;
    unsigned char *data = calloc(total,1);
    memcpy(data,msg,len);
    data[len]=0x80;

    MD5union l;
    l.w = len*8;
    memcpy(data+total-8,&l.w,4);

    for(i=0;i<total;i+=64){
        unsigned w[16],a=h[0],b=h[1],c=h[2],d=h[3],f,g,temp;
        memcpy(w,data+i,64);

        for(j=0;j<64;j++){
            if(j<16){
                f=F(b,c,d);
                g=j;
```



```c
            }
            else if(j<32){
                f=G(b,c,d);
                g=(5*j+1)%16;
            }
            else if(j<48){
                f=H(b,c,d);
                g=(3*j+5)%16;
            }
            else{
                f=I(b,c,d);
                g=(7*j)%16;
            }

            temp=d;
            d=c;
            c=b;
            b=b+rol(a+f+k[j]+w[g], (j%4==0)?7:(j%4==1)?12:(j%4==2)?17:22);
            a=temp;
        }

        h[0]+=a;
        h[1]+=b;
        h[2]+=c;
        h[3]+=d;
    }

    return h;
}

void main(){
    const char *msg="The quick brown fox jumps over the lazy dog";
    unsigned *d = md5(msg, strlen(msg));
    MD5union u;
    int i,j;

    clrscr();
    printf("MD5 ENCRYPTION ALGORITHM IN C\n\n");
    printf("Input : %s\n\n",msg);
    printf("MD5 : 0x");

    for(i=0;i<4;i++){
        u.w=d[i];
        for(j=0;j<4;j++)
            printf("%02x", u.b[j]);
    }

    printf("\n\nMD5 Encryption Successfully Completed!!!");
    getch();
}
```

OUTPUT

```
MD5 ENCRYPTION ALGORITHM IN C

Input : The quick brown fox jumps over the lazy dog

MD5 : 0x9e107d9d372bb6826bd81d3542a419d6

✓ MD5 Encryption Successfully Completed!
```

RESULT

Thus, the MD5 hashing algorithm was successfully implemented using C programming language. The implementation demonstrates the four auxiliary functions (F, G, H, I) and the 64 rounds of processing that characterize the MD5 algorithm.