18. Write a C program for DES the first 24 bits of each subkey come from the same subset of 28 bits of the initial key and that the second 24 bits of each subkey come from a disjoint subset of 28 bits of the initial key.

#include <stdio.h>

#include <string.h>

int shiftSchedule[16] = {

1, 1, 2, 2, 2, 2, 2, 2,

1, 2, 2, 2, 2, 2, 2, 1

};

int PC1[56] = {

57,49,41,33,25,17,9,

1,58,50,42,34,26,18,

10,2,59,51,43,35,27,

19,11,3,60,52,44,36,

63,55,47,39,31,23,15,

7,62,54,46,38,30,22,

14,6,61,53,45,37,29,

21,13,5,28,20,12,4

};

int PC2[48] = {

14,17,11,24,1,5,

3,28,15,6,21,10,

23,19,12,4,26,8,

16,7,27,20,13,2,

41,52,31,37,47,55,

30,40,51,45,33,48,

44,49,39,56,34,53,

46,42,50,36,29,32

};

void permute(char\* in, char\* out, int\* table, int n) {

for (int i = 0; i < n; i++)

out[i] = in[table[i] - 1];

}

void leftShift(char\* half, int shifts) {

char temp[2];

for (int s = 0; s < shifts; s++) {

temp[0] = half[0];

temp[1] = half[1];

for (int i = 0; i < 26; i++)

half[i] = half[i + 2];

half[26] = temp[0];

half[27] = temp[1];

}

}

int main() {

char key64[65], key56[56], C[28], D[28], CD[56], subkey[48];

char roundKeys[16][48];

printf("Enter 64-bit binary key (no spaces): ");

scanf("%64s", key64);

permute(key64, key56, PC1, 56);

memcpy(C, key56, 28);

memcpy(D, key56 + 28, 28);

for (int i = 0; i < 16; i++) {

leftShift(C, shiftSchedule[i]);

leftShift(D, shiftSchedule[i]);

memcpy(CD, C, 28);

memcpy(CD + 28, D, 28);

permute(CD, subkey, PC2, 48);

memcpy(roundKeys[i], subkey, 48);

}

printf("\nDES Subkeys (C-part | D-part):\n");

for (int i = 0; i < 16; i++) {

printf("K%02d: ", i + 1);

for (int j = 0; j < 48; j++) {

printf("%c", roundKeys[i][j]);

if (j == 23) printf(" | ");

}

printf("\n");

}

return 0;

}

OUTPUT:

