



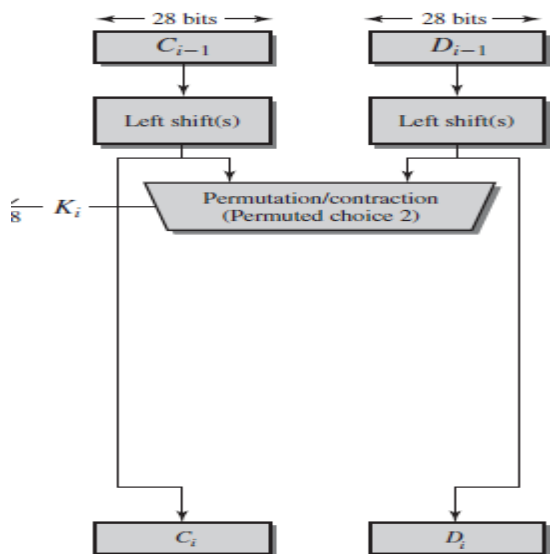
**SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMKUR-572103**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**CRYPTOGRAPHY AND NETWORK SECURITY LAB (7RCSL01)**

Student Name: Pragati Shankar		USN: 1SI19CS090	Batch No: B2	Date: 06-12-2022
<b>Evaluation:</b>				
Write Up (10 marks)	Clarity in concepts (10 marks)	Implementation and execution of the algorithms (10 marks)	Viva (05 marks)	Total (35 marks)
Sl.No	Name of the Faculty In-Charge			Signature
1.	H K Vedamurthy			
2.	Gururaj S P			

**Question No: 5**

Generate and print 48-bit keys for all sixteen rounds of DES algorithm, given a 64-bit initial key.

Algorithm: To Generate 48-bits key, follow the flow-chart and tables given below.



(a) Input Key

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

(b) Permuted Choice One (PC-1)

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

(c) Permuted Choice Two (PC-2)

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

(d) Schedule of Left Shifts

Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits Rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Figure: DES key Schedule Calculation

Tables: DES key Schedule Calculation

## CODE:

```
#include <bits/stdc++.h>
using namespace std;

int permChoiceOne[] = {
    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 permChoiceTwo[] = {
    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 };

int leftShiftTable[] = {1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1};

string rotateSubKey(string s , int rot)
{
    return s.substr(rot, s.length()-rot) + s.substr(0, rot) ;
}

string firstPermute(string input)
{
    string res = "" ;
    for(int i=0 ; i<56 ; i++)
    {
        res += input[permChoiceOne[i]-1];
    }
    return res ;
}

string secondPermute(string input)
{
    string res = "" ;
    for(int i=0 ; i<48 ; i++)
    {
```

```

        res += input[permChoiceTwo[i]-1];
    }
    return res ;
}

void genKeys(string left, string right)
{
    ofstream fout ;
    fout.open("keygen.txt");
    for (int i=0; i<16; i++)
    {
        left = rotateSubKey(left , leftShiftTable[i]);
        right = rotateSubKey(right, leftShiftTable[i]);
        string key = secondPermute(left+right);
        cout << "key " << i+1 << " \t: " << key << endl;
        fout << key << endl;
    }
}

int main()
{
    unsigned long long hexkey;
    cout << "\nEnter 64-bit key in hexadecimal(16-digits) : " ;
    cin >> hex >> hexkey;

    string key = bitset<64>(hexkey).to_string();
    cout << "Binary key (k) \t: " << key << endl;

    key = firstPermute(key) ;
    cout << "PC-1 key (k+) \t: " << key << endl;

    cout << "\nSubKeys: " << endl;
    genKeys(key.substr(0,28) , key.substr(28,28));

    cout<<endl<<endl ;

    return 0;
}

```

## Output Screenshot:

```
user@linux-OptiPlex-5090: ~/1SI19CS090
user@linux-OptiPlex-5090:~/1SI19CS090$ g++ des1.cpp
user@linux-OptiPlex-5090:~/1SI19CS090$ ./a.out

Enter 64-bit key in hexadecimal(16-digits) : 1FE22472901BB2A3
Binary key (k) : 0001111111100010001001000111001010010000000110111011001010100011
PC-1 key (k+) : 11010010000010101100111001111110101100000101001000011001

SubKeys:
key 1 : 1100101101100010101100110001101000000110100011
key 2 : 001101011010010010101111001001011110000110011010
key 3 : 111100110000110010000010011001010001010001000011
key 4 : 011110001010101010110100110011101000000001101110
key 5 : 10010100101101000001111000000100110111111001100
key 6 : 011001100000011001110110000110001001010011110001
key 7 : 111011101101100000100100110010111100110000100001
key 8 : 1000101010001101111010000010100110111100011000
key 9 : 111000111100001011010111100101000001110000110011
key 10 : 001111011101001110000010100011110000101001110100
key 11 : 001100100001000111111011000100011110101111010000
key 12 : 101111010100000001010101001100011000010000010101
key 13 : 000001110100101110011100110010110010010010000110
key 14 : 000111100011000110110101001011000110001110001101
key 15 : 100111110000110001101001001100100101000011000111
key 16 : 010010011010100110011011011010100010100101010110

user@linux-OptiPlex-5090:~/1SI19CS090$
```

```
keygen.txt
~/1SI19CS090

1 1100101101100010101100110001101000000110100011
2 001101011010010010101111001001011110000110011010
3 111100110000110010000010011001010001010001000011
4 011110001010101010110100110011101000000001101110
5 10010100101101000001111000000100110111111001100
6 011001100000011001110110000110001001010011110001
7 111011101101100000100100110010111100110000100001
8 1000101010001101111010000010100110111100011000
9 111000111100001011010111100101000001110000110011
10 001111011101001110000010100011110000101001110100
11 001100100001000111111011000100011110101111010000
12 101111010100000001010101001100011000010000010101
13 000001110100101110011100110010110010010010000110
14 000111100011000110110101001011000110001110001101
15 100111110000110001101001001100100101000011000111
16 010010011010100110011011011010100010100101010110

Plain Text ▾ Tab Width: 8 ▾ Ln 14, Col 49 ▾ INS
```