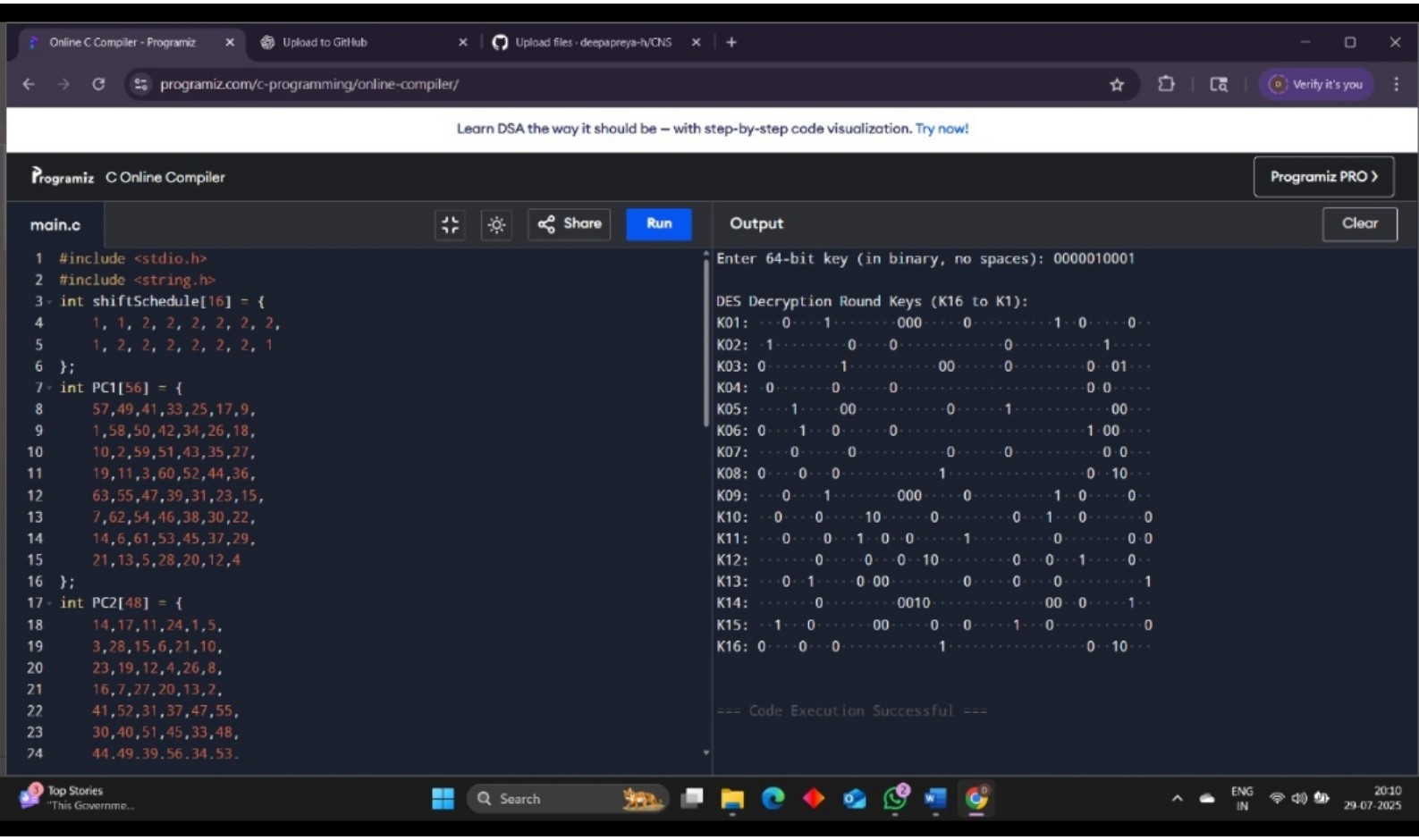
Online C Compiler - Programiz
Upload to GitHub
Upload files · deepapreya-h/CNS
programiz.com/c-programming/online-compiler/
Verify it's you
Learn DSA the way it should be — with step-by-step code visualization. Try now!
Programiz C Online Compiler
Programiz PRO
main.c
Share
Run
Output
Clear
#include <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,
Enter 64-bit key (in binary, no spaces): 0000010001
DES Decryption Round Keys (K16 to K1):
K01:
K02:
K03:
K04:
K05:
K06:
K07:
K08:
K09:
K10:
K11:
K12:
K13:
K14:
K15:
K16:
=== Code Execution Successful ===
Top Stories
"This Governme...
Search
ENG
IN
20:10
29-07-2025

Online C Compiler - Programiz | Upload to GitHub | Upload files · deepapreya-h/CNS

programiz.com/c-programming/online-compiler/

Learn DSA the way it should be — with step-by-step code visualization. Try now!

Programiz C Online Compiler | Programiz PRO >

main.c | Share | Run

```c
    30,40,51,45,33,48,
    44,49,39,56,34,53,
    46,42,50,36,29,32
};
void permute(char *input, char *output, int *table, int size) {
    for (int i = 0; i < size; i++) {
        output[i] = input[table[i] - 1];
    }
}
void leftShift(char *key_half, int shifts) {
    char temp[2];
    for (int s = 0; s < shifts; s++) {
        temp[0] = key_half[0];
        temp[1] = key_half[1];
        for (int i = 0; i < 26; i++)
            key_half[i] = key_half[i + 2];
        key_half[26] = temp[0];
        key_half[27] = temp[1];
    }
}
int main() {
    char initialKey[64];
    char permutedKey[56];
    char C[28], D[28], CD[56], roundKey[48];
```

Output | Clear

```
Enter 64-bit key (in binary, no spaces): 0000010001

DES Decryption Round Keys (K16 to K1):
K01: ···0····1·······000····0···········1··0·····0··
K02: ·1·········0····0·············0············1·····
K03: 0·········1···········00·····0··········0··01···
K04: ·0·······0······0···················0·0·····
K05: ····1····00············0······1············00···
K06: 0····1···0······0·····················1·00···
K07: ····0······0············0······0···········0·0···
K08: 0····0···0············1·················0··10···
K09: ···0····1·······000····0···········1··0·····0··
K10: ··0····0·····10·····0·········0···1···0·······0
K11: ···0····0···1··0··0······1··········0·······0·0
K12: ·······0·····0···0··10··········0···0···1·····0··
K13: ···0··1····0·00········0·····0····0············1
K14: ·······0·········0010··············00··0·····1··
K15: ··1···0·······00·····0···0·····1···0············0
K16: 0····0···0············1·················0··10···


=== Code Execution Successful ===
```

Top Stories "This Governme... Search ENG IN 20:10 29-07-2025

```c
    char permutedKey[56];
    char C[28], D[28], CD[56], roundKey[48];
    char roundKeys[16][48];
    printf("Enter 64-bit key (in binary, no spaces): ");
    scanf("%64s", initialKey);
    permute(initialKey, permutedKey, PC1, 56);
    memcpy(C, permutedKey, 28);
    memcpy(D, permutedKey + 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, roundKeys[i], PC2, 48);
    }
    printf("\nDES Decryption Round Keys (K16 to K1):\n");
    for (int i = 15; i >= 0; i--) {
        printf("K%02d: ", 16 - i);
        for (int j = 0; j < 48; j++)
            printf("%c", roundKeys[i][j]);
        printf("\n");
    }
    return 0;
}
```

Output

```
Enter 64-bit key (in binary, no spaces): 0000010001

DES Decryption Round Keys (K16 to K1):
K01: 0 1 000 0 1 0 0
K02: 1 0 0 0 1
K03: 0 1 00 0 0 01
K04: 0 0 0 0 0
K05: 1 00 0 1 00
K06: 0 1 0 0 1 00
K07: 0 0 0 0 0 0
K08: 0 0 0 1 0 10
K09: 0 1 000 0 1 0 0
K10: 0 0 10 0 0 1 0 0
K11: 0 0 1 0 0 1 0 0 0
K12: 0 0 0 10 0 0 1 0
K13: 0 1 0 00 0 0 0 1
K14: 0 0010 00 0 1
K15: 1 0 00 0 0 1 0 0
K16: 0 0 0 1 0 10

=== Code Execution Successful ===
```