

Output :- Modular Arithmetic

```
cd "/Users/Don't Open/5th Sem/Lab/Aryan_28900/Crypto/Lab/Lab2/" && gcc Modular_Arithmetic.c
(base) aryankushwaha@Aryan-Kushwaha Lab2 % cd "/Users/Don't Open/5th Sem/Lab/Aryan_28900/C
sers/Don't Open/5th Sem/Lab/Aryan_28900/Crypto/Lab/Lab2/"Modular_Arithmetic

Menu:
1. Find Additive Inverse
2. Find Multiplicative Inverse (Extended Euclidean Algorithm)
3. Check if Two Numbers are Relatively Prime
4. Exit
Enter your choice: 1
Enter a number: 34
Enter the modulo: 13
The additive inverse of 34 under modulo 13 is: 5

Menu:
1. Find Additive Inverse
2. Find Multiplicative Inverse (Extended Euclidean Algorithm)
3. Check if Two Numbers are Relatively Prime
4. Exit
Enter your choice: 2
Enter a number: 45
Enter the modulo: 17
The multiplicative inverse of 45 under modulo 17 is: 14

Menu:
1. Find Additive Inverse
2. Find Multiplicative Inverse (Extended Euclidean Algorithm)
3. Check if Two Numbers are Relatively Prime
4. Exit
Enter your choice: 3
Enter the first number: 17
Enter the second number: 19
17 and 19 are relatively prime.

Menu:
1. Find Additive Inverse
2. Find Multiplicative Inverse (Extended Euclidean Algorithm)
3. Check if Two Numbers are Relatively Prime
4. Exit
Enter your choice: 4
Exiting...
(base) aryankushwaha@Aryan-Kushwaha Lab2 %
```

Output : - S-box Implementation for AES

```
cd "/Users/Don't Open/5th Sem/Lab/Aryan_28900/Crypto/Lab/Lab2/" && gcc S_Box_AES.c -o S_Box_AES && "/Users/Don't Open/5th Sem/Lab/Aryan_28900/Crypto/Lab/Lab2/" && gcc S_Box_AES
(base) aryankushwaha@Aryan-Kushwaha Lab2 % cd "/Users/Don't Open/5th Sem/Lab/Aryan_28900/Crypto/Lab/Lab2/" && gcc S_Box_AES

Menu:
1. Substitute byte using S-box
2. Substitute byte using Inverse S-box
3. Exit
Enter your choice: 1
Enter a byte (in hex, e.g., 53): F5
Input byte: 0xf5
Int value for the input is 245
Substituted byte: 0xe6

Menu:
1. Substitute byte using S-box
2. Substitute byte using Inverse S-box
3. Exit
Enter your choice: 2
Enter a byte (in hex, e.g., 53): A0
Input byte: 0xa0
Inverse substituted byte: 0x47

Menu:
1. Substitute byte using S-box
2. Substitute byte using Inverse S-box
3. Exit
Enter your choice: 3
Exiting...
(base) aryankushwaha@Aryan-Kushwaha Lab2 %
```

Output : - Key Generation Process in DES

```
cd "/Users/Don't Open/5th Sem/Lab/Aryan_28900/Crypto/Lab/Lab2/" && gcc S_Box_DES.c -o S_Box_DES
● (base) aryankushwaha@Aryan-Kushwaha Lab2 % cd "/Users/Don't Open/5th Sem/Lab/Aryan_28900/Crypto/Lab/Lab2/"S_Box_DES
Subkey 1: 00110111010111111111001101000000000100100000100
Subkey 2: 011111101111101100011001000001001000000001000000
Subkey 3: 00001111011110101111111000000001010010001000000
Subkey 4: 111011110110110011011111001010001000011000000000
Subkey 5: 011111111110111110101000000110000100010000000010
Subkey 6: 110110101011110110111011000011000100000000000000
Subkey 7: 111111011010111001011111100000000110000001000000
Subkey 8: 011001111111111010001110101000001000001000000000
Subkey 9: 111101000011011111101101000000110000000000011000
Subkey 10: 110100111101111001110101000000010001000100000100
Subkey 11: 110011011111101111110110000000000000000010100100
Subkey 12: 101101101111011111101111010000000000100010000101
Subkey 13: 111110110101011101100011000000100000000010011001
Subkey 14: 111010011101101111111101000000110001000100000001
Subkey 15: 100101011111001111011111000000100000000100100000
Subkey 16: 111111100100111111011110000101000100000001000010
○ (base) aryankushwaha@Aryan-Kushwaha Lab2 %
```