Write a C Program to implement Shift Cipher.

Program:

```
#include <stdio.h>
#include <ctype.h>
int main() {
   char st[100];
   int key, i;
   char ch, m;
   printf("Enter the plain text: \n");
   scanf("%s", st);
   printf("The plain text is %s\n", st);
   printf("Enter the key: \n");
   scanf("%d", &key);
   for (i = 0; st[i] != '\0'; i++) {
      ch = st[i];
      if (isalnum(ch)) {
         if (islower(ch)) {
            ch = (ch - 'a' + key) % 26 + 'a';
         }
         if (isupper(ch)) {
            ch = (ch - 'A' + key) % 26 + 'A';
         }
         if (isdigit(ch)) {
            ch = (ch - '0' + key) % 10 + '0';
         }
      } else {
         printf("Invalid character\n");
      }
      st[i] = ch;
   }
   printf("\nThe encrypted text is: %s\n", st);
   for (i = 0; st[i] != '\0'; i++) {
      m = st[i];
      if (isalnum(m)) {
         if (islower(m)) {
            m = (m - 'a' - key) % 26 + 'a';
         }
         if (isupper(m)) {
            m = (m - 'A' - key) % 26 + 'A';
         }
         if (isdigit(m)) {
            m = (m - '0' - key) % 10 + '0';
         }
      } else {
         printf("Invalid character\n");
      }
```

```
        st[i] = m;
    }
    printf("\nThe decrypted text is: %s\n", st);
}
```

Output:

```
Enter the plain text:
hello
The plain text is hello
Enter the key:
3

The encrypted text is: khoor

The decrypted text is: hello
```

Write a C Program to implement Mono-Alphabetic Substitution Cipher.

Program:

```
#include<stdio.h>
char monocipher_encr(char);
char monocipher_deencr(char);
char alpha[27][3] = { { 'a', 'f' }, { 'b', 'a' }, { 'c', 'g' }, { 'd', 'u' }, {'e', 'n' }, { 'f', 'i' }, { 'g', 'j' }, { 'h', 'k' }, { 'i', 'l' },
{'j', 'm' }, { 'k', 'o' }, { 'l', 'p' }, { 'm', 'q' }, { 'n', 'r' }, {'o', 's' }, { 'p', 't' }, { 'q', 'v' }, { 'r', 'w' }, { 's', 'x' }, {'t', 'y' }, {'v',
'b' }, { 'u', 'z' }, { 'w', 'c' }, { 'x', 'd' }, {'y', 'e' }, { 'z', 'h' } };
char str[20];
char str3[20];
int main() {
    char str[20], str2[20];
    int i, j;
    printf("Enter string: ");
    scanf("%s", str);
    for(i = 0; i < str[i] != '\0'; i++) {
        str2[i] = monocipher_encr(str[i]);
    }
    str[i] = '\0';
    printf("Before encryption : %s", str);
    printf("\nAfter encryption : %s", str2);
    for(j = 0; j < str2[j] != '\0'; j++) {
        str3[j] = monocipher_deencr(str2[j]);
    }
    str3[j] = '\0';
    printf("\nAfter decryption : %s", str3);
}
char monocipher_encr(char a) {
    int i;
    for(i = 0; i < 26; i++) {
        if(a == alpha[i][0]) {
```

```
            break;
        }
    }
    return alpha[i][1];
}
char monocipher_deencr(char a) {
    int i;
    for(i = 0; i < 26; i++) {
        if(a == alpha[i][1]) {
            break;
        }
    }
    return alpha[i][0];
}
```

Output:

```
Enter string: hello
Before encryption : hello
After encryption : knpps
After decryption : hello
```

Write a C Program to implement Diffie-Helman Key Exchange Algorithm.

Program:

```
#include<stdio.h>
#include<math.h>
long long int power(long long int a, long long int b, long long int P) {
    if(b == 1) return a;
    else return (((long long int) pow(a, b)) % P);
}
int main() {
    long long int P, G, x, a, y, b, ka, kb;
    P = 23;
    printf("The value of P: %lld\n", P);
    G = 9;
    printf("The value of G: %lld\n\n", G);
    a = 4;
    printf("The private key a for Alice: %lld\n", a);
    x = power(G, a, P);
    b = 3;
    printf("The private key b for Bob: %lld\n\n", b);
    y = power(G, b, P);
    ka = power(y, a, P);
    kb = power(x, b, P);
    printf("Secret key for the Alice is : %lld\n", ka);
    printf("Secret key for the Bob is : %lld\n", kb);
}
```

Output:

```
The value of P: 23
The value of G: 9

The private key a for Alice: 4
The private key b for Bob: 3

Secret key for the Alice is : 9
Secret key for the Bob is : 9
```

Write a C Program to implement Eucledian Algorithm to find GCD.

Program:

```
#include<stdio.h>
void main() {
   int m, n;
   printf("Enter two integer numbers: ");
   scanf("%d %d", &m, &n);
   while(n > 0) {
      int r = m % n;
      m = n;
      n = r;
   }
   printf("GCD = %d", m);
}
```

Output:

```
Enter two integer numbers: 25 5
GCD = 5
```