MBM
Madan Bhandari Memorial College

Lab 10: WAP for Euclidean GCD

Code:

```
#include <iostream>
using namespace std;
int gcd(int a, int b){
   if (a == 0)
      return b;
   return gcd(b % a, a);
}
int main(){
  do{
      int a, b;
      cout << "Enter the value of a: ";
      cin >> a;
      cout << "Enter the value of b: ";
      cin >> b;
      cout << "GCD(" << a << ", " << b << ") = " << gcd(a, b) << endl;

      cout << "Do you want to continue? (y/n): ";
      char choice;
      cin >> choice;
      if (choice != 'y' && choice != 'Y')
         break;
      cin.ignore();
   } while (true);
   cout << "\nLab No.6\nName: Sabin Sapkota\nRoll no: 12 \n";
   return 0;
}
```

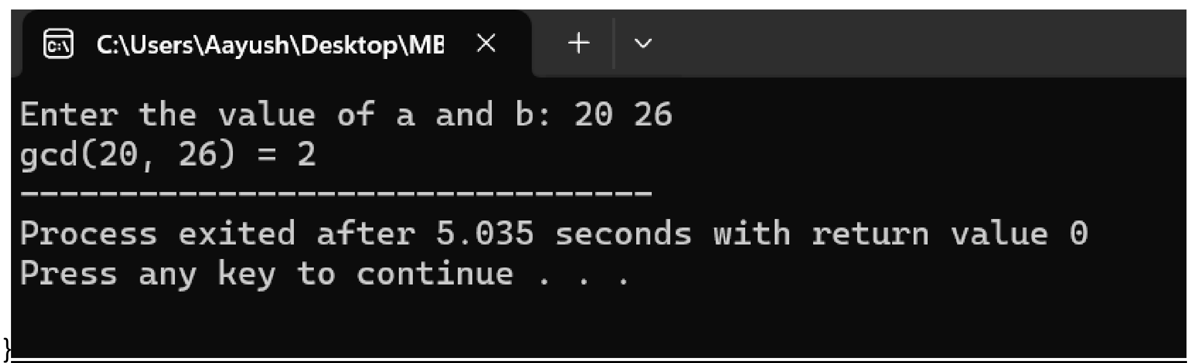
Output

```
C:\Users\Aayush\Downloads\

Enter the value of a: 15
Enter the value of b: 21
GCD(15, 21) = 3
Do you want to continue? (y/n): y
Enter the value of a: 9
Enter the value of b: 6
GCD(9, 6) = 3
Do you want to continue? (y/n): n

Lab No.10
Name: Aayush Basnet
Symbol no: 29131

--------------------------------
Process exited after 17.11 seconds with return value 0
Press any key to continue . . .
```

Lab 11: WAP for Extended Euclidean GCD

Code:

```
#include <stdio.h>
int gcdExtended(int a, int b, int *x, int *y){

    if (a == 0){
        *x = 0;
        *y = 1;
        return b;
    }

    int x1, y1;
    int gcd = gcdExtended(b%a, a, &x1, &y1);
    *x = y1 - (b/a) * x1;
    *y = x1;

    return gcd;
}

int main(){

    int x, y;
    int a, b;
    printf("Enter the value of a and b: ");
    scanf("%d %d",&a,&b);

    int g = gcdExtended(a, b, &x, &y);
    printf("gcd(%d, %d) = %d", a, b, g);

    return 0;
}
```



}

Madan Bhandari Memorial College

Lab12: WAP to write Robin Miller

Code:

```
#include <iostream>
#include <stdlib.h>
using namespace std;

long long mulmod(long long, long long, long long);
long long modulo(long long, long long, long long);
bool Miller(long long, int);

int main(){

do{
    int iteration = 10;
    long long num;
    cout << "Enter integer to test primality: ";
    cin >> num;
    if (Miller(num, iteration))
      cout << num << " is prime" << endl;
    else
      cout << num << " is not prime" << endl;
    char choice;
    cout << "Do you want to continue? (y/n): ";
    cin >> choice;
    if (choice == 'n' || choice == 'N')
      break;
  } while (true);
  cin.get();
 return 0;
}
long long mulmod(long long a, long long b, long long m){
  long long x = 0,
       y = a % m;
  while (b > 0){
    if (b % 2 == 1){
      x = (x + y) % m;
    }

    y = (y * 2) % m;
    b /= 2;
  }
  return x % m;
}

long long modulo(long long base, long long e, long long m){
  long long x = 1;
  long long y = base;

 while (e > 0){
         if (e % 2 == 1)
      x = (x * y) % m;
```

```
        y = (y * y) % m;
        e = e / 2;
    }
    return x % m;
}
bool Miller(long long p, int iteration){
    if (p < 2){
        return false;
    }
    if (p != 2 && p % 2 == 0){
        return false;
    }

    long long s = p - 1;
    while (s % 2 == 0){
        s /= 2;
    }

    for (int i = 0; i < iteration; i++){
        long long a = rand() % (p - 1) + 1, temp = s;
        long long mod = modulo(a, temp, p);
        while (temp != p - 1 && mod != 1 && mod != p - 1){
            mod = mulmod(mod, mod, p);
            temp *= 2;
        }
        if (mod != p - 1 && temp % 2 == 0){
            return false;
        }
    }

    return true;
}
```

```
C:\Users\Aayush\Downloads\

Enter integer to test primality: 49
49 is not prime
Do you want to continue? (y/n): y
Enter integer to test primality: 19
19 is prime
Do you want to continue? (y/n): n

--------------------------------
Process exited after 7.711 seconds with return value 0
Press any key to continue . . .
```

Madan Bhandari Memorial College

Lab 13: WAP to calculate S-Box Substitution

Code:

```
#include <iostream>
#include <bitset>
#include <string>
using namespace std;

const int S1[4][16] = {
   {14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7},
   {0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8},
   {4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0},
   {15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13}};

bitset<4> S1Substitution(bitset<6> input){
   int row = (input[5] << 1) + input[0];
   int col = (input[4] << 3) + (input[3] << 2) + (input[2] << 1) + input[1];
   int output = S1[row][col];
   return bitset<4>(output);
}
int main(){
   do{
      string inputStr;
      cout << "Enter a 6-bit binary input: ";
      cin >> inputStr;
                  if (inputStr.length() != 6){
         cout << "Input must be 6 bits long." << endl;
         return 1;
      }

      bitset<6> input(inputStr);

      bitset<4> output = S1Substitution(input);

      cout << "Input:  " << input << endl;
      cout << "Output: " << output << endl;
      char choice;
      cout << "Do you want to continue? (y/n): ";
      cin >> choice;
      if (choice == 'n' || choice == 'N')
         break;
   } while (true);
   return 0;
}
```

Output

C:\Users\Aayush\Downloads\

```
Enter a 6-bit binary input: 100110
Input:  100110
Output: 1000
Do you want to continue? (y/n): y
Enter a 6-bit binary input: 1011
Input must be 6 bits long.

--------------------------------
Process exited after 12.38 seconds with return value 1
Press any key to continue . . .
```

Lab 15: WAP to calculate Discrete Logarithm

Code:

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

int discreteLogarithm(int a, int b, int m) {
  int n = (int) sqrt (m) + 1;
  int an = 1;
  for (int i = 0; i<n; ++i)
    an = (an * a) % m;

  for (int i = 1, cur = an; i<= n; ++i) {
    if (! value[ cur ])
      value[ cur ] = i;
    cur = (cur * an) % m;
  }

  for (int i = 0, cur = b; i<= n; ++i){

    if (value[cur]){
      int ans = value[cur] * n - i;
      if (ans < m)
        return ans;
    }
    cur = (cur * a) % m;
  }
  return -1;
}


int main(){
  int a = 2, b = 3, m = 5;
  cout << discreteLogarithm(a, b, m) << endl;

  a = 3, b = 7, m = 11;
  cout << discreteLogarithm(a, b, m);
}
```

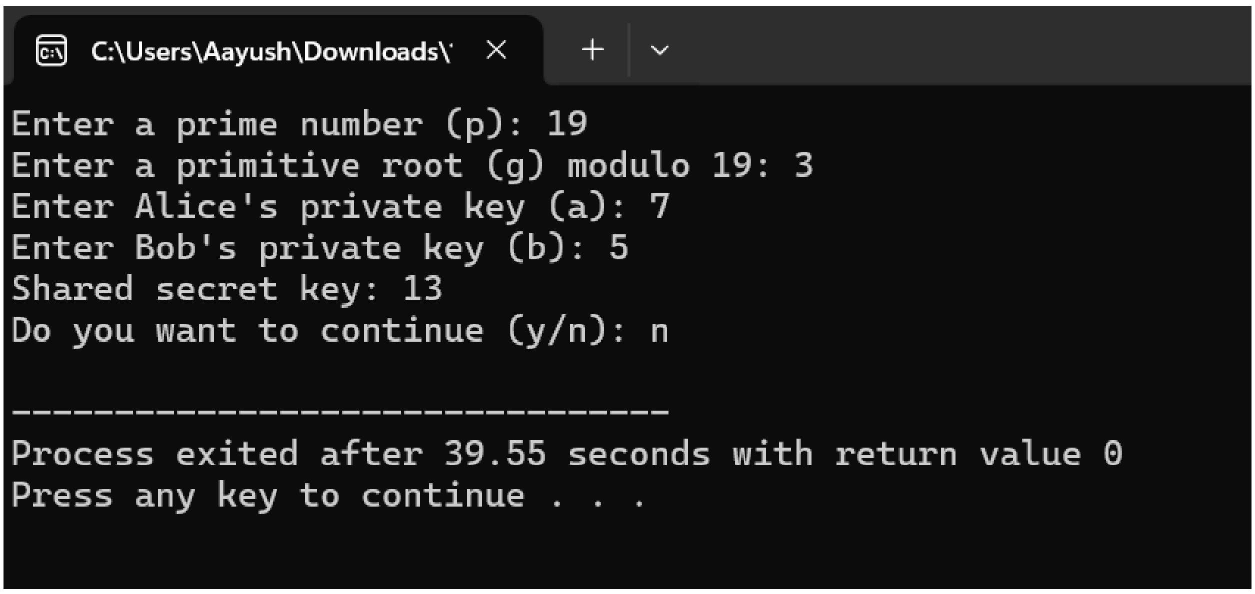
Lab 16: WAP to calculate the Key for two persons using the Diffie-Hellman Key exchange algorithm

Code:

```
#include <iostream>
#include <cmath>
using namespace std;

int modExp(int base, int exponent, int modulus){
    if (exponent == 0)
        return 1;
    int result = 1;
    base = base % modulus;

 while (exponent > 0){
        if (exponent % 2 == 1){
            result = (result * base) % modulus;
        }
        exponent = exponent >> 1; // Right shift exponent by 1
        base = (base * base) % modulus;
    }

    return result;
}

int main(){

    int p, g, a, b;

 do {
        cout << "Enter a prime number (p): ";
        cin >> p;
        cout << "Enter a primitive root (g) modulo " << p << ": ";
        cin >> g;
        cout << "Enter Alice's private key (a): ";
        cin >> a;
        cout << "Enter Bob's private key (b): ";
        cin >> b;
        int A = modExp(g, a, p);  // Calculate Alice's public key
        int B = modExp(g, b, p);  // Calculate Bob's public key
        int s1 = modExp(B, a, p); // Calculate shared secret key for Alice
        int s2 = modExp(A, b, p); // Calculate shared secret key for Bob

  if (s1 == s2)
            cout << "Shared secret key: " << s1 << endl;
        else
            cout << "Key exchange failed!" << endl;

         char choice;
        cout << "Do you want to continue (y/n): ";
        cin >> choice;
```

```
        if (choice == 'n')
            break;
    } while (true);
    cin.get();

    return 0;
}
```

Output




```
Enter a prime number (p): 19
Enter a primitive root (g) modulo 19: 3
Enter Alice's private key (a): 7
Enter Bob's private key (b): 5
Shared secret key: 13
Do you want to continue (y/n): n

--------------------------------
Process exited after 39.55 seconds with return value 0
Press any key to continue . . .
```

Lab 17: Write a program for RSA asymmetric cryptographic algorithm.

Code:

```
#include <iostream>
#include <cmath>
using namespace std;

long long mod_pow(long long a, long long b, long long c){
   long long result = 1;
   a = a % c;
   while (b > 0){
      if (b % 2 == 1){
         result = (result * a) % c;
      }
      a = (a * a) % c;
      b /= 2;
   }
   return result;
}

long long encrypt(long long message, long long e, long long n){
   return mod_pow(message, e, n);
}
long long decrypt(long long encrypted, long long d, long long n){
   return mod_pow(encrypted, d, n);
}

int main(){
   long long p, q, n, phi, e, d;
   long long message, encrypted, decrypted;
   char choice;
   cout << "RSA Encryption and Decryption Menu" << endl;

 do{
      cout << "1. Key Generation\n2. Encrypt\n3. Decrypt\n4. Exit" << endl;
      cout << "Enter your choice: ";
      cin >> choice;

  switch (choice)  {
      case '1':
         cout << "Enter two prime numbers (p and q): ";
         cin >> p >> q;
         n = p * q;
         phi = (p - 1) * (q - 1);
         cout << "Enter a public key (e): ";
         cin >> e;
         d = 1;
         while ((d * e) % phi != 1){
            d++;
         }
```

```cpp
            cout << "Keys generated: " << endl;
            cout << "Public key (e, n): (" << e << ", " << n << ")" << endl;
            cout << "Private key (d, n): (" << d << ", " << n << ")" << endl;
            break;

        case '2':
            cout << "Enter the message to encrypt: ";
            cin >> message;
            encrypted = encrypt(message, e, n);
            cout << "Encrypted message: " << encrypted << endl;
            break;

        case '3':
            cout << "Enter the message to decrypt: ";
            cin >> encrypted;
            decrypted = decrypt(encrypted, d, n);
            cout << "Decrypted message: " << decrypted << endl;
            break;

        case '4':
            cout << "Exiting the program. Goodbye!" << endl;
            break;

        default:
            cout << "Invalid choice. Please try again." << endl;
        }
    } while (choice != '4');

    return 0;
}
```

C:\Users\Aayush\Desktop\ME

```
RSA Encryption and Decryption Menu
1. Key Generation
2. Encrypt
3. Decrypt
4. Exit
Enter your choice: 1
Enter two prime numbers (p and q): 53 59
Enter a public key (e): 3127
Keys generated:
Public key (e, n): (3127, 3127)
Private key (d, n): (951, 3127)
1. Key Generation
2. Encrypt
3. Decrypt
4. Exit
Enter your choice: 4
Exiting the program. Goodbye!

--------------------------------
Process exited after 119.4 seconds with return value 0
Press any key to continue . . .
```

Madan Bhandari Memorial College

Lab 18: Write a program to illustrate ElGamal encryption

Code:

```
#include <iostream>
#include <cmath>
#include <cstdlib>
#include <ctime>
using namespace std;
long long mod_pow(long long a, long long b, long long c){
    long long result = 1;
    a = a % c;
    while (b > 0){
        if (b % 2 == 1){
            result = (result * a) % c;
        }
        a = (a * a) % c;
        b /= 2;
    }
    return result;
}

long long generate_random_prime(){
    long long n = rand() % 1000 + 1000; // Generate a random number in a certain range
    for (long long i = n;; i++)  {
        bool is_prime = true;
        for (long long j = 2; j <= sqrt(i); j++) {
            if (i % j == 0){
                is_prime = false;
                break;
            }
        }

        if (is_prime){
            return i;
        }
    }
}

int main(){
    srand(static_cast<unsigned>(time(0)));
    long long p, g, x, y, k, m, a, b;
    long long decrypted_m; // Declare decrypted_m here
    char choice;
    do{
        cout << "ElGamal Cryptographic System Menu" << endl;
        cout << "1. Key Generation\n2. Encrypt\n3. Decrypt\n4. Exit" << endl;
        cout << "Enter your choice: ";
        cin >> choice;
```

```
switch (choice){
 case '1':
        p = generate_random_prime();
        g = generate_random_prime();
        cout << "Enter your private key x: ";
        cin >> x;
        y = mod_pow(g, x, p);
        cout << "Keys generated: " << endl;
        cout << "Public Key (p, g, y): (" << p << ", " << g << ", " << y << ")" << endl;
        break;

      case '2':
        cout << "Enter the message to encrypt (an integer): ";
        cin >> m;
        k = rand() % (p - 2) + 1; // Random value in the range [1, p-1]
        a = mod_pow(g, k, p);
        b = (m * mod_pow(y, k, p)) % p;
        cout << "Ciphertext (a, b): (" << a << ", " << b << ")" << endl;
        break;

      case '3':
        decrypted_m = (b * mod_pow(a, p - 1 - x, p)) % p;
        cout << "Decrypted Message: " << decrypted_m << endl;
        break;

      case '4':
        cout << "Exiting the program. Goodbye!" << endl;
        break;

      default:
        cout << "Invalid choice. Please try again." << endl;
      }
    } while (choice != '4');

  return 0;
}
```

Output

```
C:\Users\Aayush\Downloads\

ElGamal Cryptographic System Menu
1. Key Generation
2. Encrypt
3. Decrypt
4. Exit
Enter your choice: 1
Enter your private key x: 5
Keys generated:
Public Key (p, g, y): (1087, 1193, 942)
ElGamal Cryptographic System Menu
1. Key Generation
2. Encrypt
3. Decrypt
4. Exit
Enter your choice: 2
Enter the message to encrypt (an integer): 45
Ciphertext (a, b): (214, 901)
ElGamal Cryptographic System Menu
1. Key Generation
2. Encrypt
3. Decrypt
4. Exit
Enter your choice: 4
Exiting the program. Goodbye!

--------------------------------
Process exited after 21.05 seconds with return value 0
Press any key to continue . . .
```

Name: Aayush Basnet **Date:**

Lab 6: WAP to find additive inverse.

Theory:

Additive inverse is the number that is added to a given number to make the sum zero. For example, if we take the number 3 and add -3 to it, the result is zero. Hence, the additive inverse of 3 is -3. We come across such situations in our daily life where we nullify the value of a quantity by taking its additive inverse.

The additive inverse of a number is its opposite number. If a number is added to its additive inverse, the sum of both the numbers becomes zero. The simple rule is to change the positive number to a negative number and vice versa. We know that, 7+ (-7)

=0. Thus -7 is the additive inverse of 7 and 7 is the additive inverse of -7.

0	0
1	7
2	6
3	5
4	4
5	3
6	2
7	1

Additive inverse modulo 8

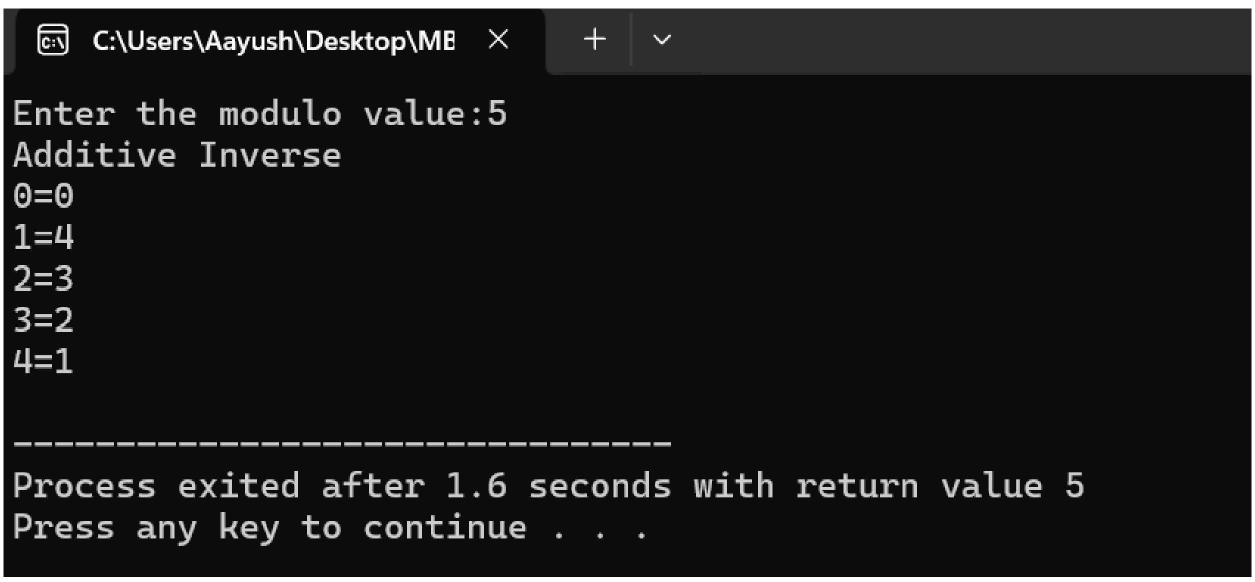
Programming Language: C

IDE: Dev C++

Code :

```
#include <stdio.h>

int main()
{
   int i, n, inv, m;

   printf("Enter the modulo value:");
   scanf("%d", &m);

   printf("Additive Inverse\n");

   for (i = 0; i < m; i++)
   {
      printf("%d=%d\n", i, inv = (m - i) % m);
   }
}
```

Output



Madan Bhandari Memorial College

Name: Aayush Basnet **Date:**

Lab 7: WAP to find multiplicative inverse.

Theory:

The meaning of the word “inverse” is something opposite in effect. The multiplicative inverse of a number is a number that, when multiplied by the given number, gives 1 as the product. By multiplicative inverse definition, it is the reciprocal of a number. The multiplicative inverse of a number “a” is represented as a-1 or (1/a).

The multiplicative inverse property states that if we multiply a number with its reciprocal, the product is always equal to 1. The image given below shows that (1/a) is the reciprocal of the number “a”.

A pair of numbers, when multiplied to give product 1, are said to be multiplicative inverses of each other. Here, a and (1/a) are reciprocals of each other.

0	-
1	1
2	-
3	3
4	-
5	5
6	-
7	7

Multiplicative inverse modulo 8

Programming Language: C

IDE: Dev C++

Code :

```
#include <stdio.h>
int gcd(int n1, int n2)
{
    if (n2 != 0)
        return gcd(n2, n1 % n2);
    else
        return n1;
}
void main()
{
    int s, m, i, num, MI, j;
    printf("Enter the modulo value:\n");
    scanf("%d", &m);
    for (j = 0; j < m; j++)
    {
        if (gcd(j, m) == 1)
        {
            for (i = 1; i <= j; i++)
            {
                s = ((i * m) + i);
                MI = s % m;
                if (MI % j == 0)
                {
                    break;
                }
            }
            printf("Multiplicative inverse of %d is %d\n", j, MI);
        }
        else
        {
            printf("Multiplicative inverse of %d can not be calculated\n", j);
        }
    }
}
```

Output

C:\Users\Aayush\Desktop\MB

```
Enter the modulo value:
6
Multiplicative inverse of 0 can not be calculated
Multiplicative inverse of 1 is 1
Multiplicative inverse of 2 can not be calculated
Multiplicative inverse of 3 can not be calculated
Multiplicative inverse of 4 can not be calculated
Multiplicative inverse of 5 is 5

--------------------------------
Process exited after 1.396 seconds with return value 6
Press any key to continue . . .
```

Name: Aayush Basnet **Date:**

Lab 8: WAP for totient function

Theory:

Totient function also known as Euler's Totient function can be defined as the number of positive integer less than n, which are relatively prime to n. It is denoted by $\Phi(n)$.

Example: $\Phi(10)$=?

Here, n = 10

Numbers less than 10 are: {1,2,3,4,5,6,7,8,9}

Numbers relatively prime to 10 are: {1,3,7,9}

$\therefore \Phi(10) = 4$

If n is the prime number then $\Phi(n) = n-1$

Example: $\Phi(7) = ?$

$\Phi(7) = 7-1$

$= 6$

Let p and q be the two prime numbers such that $p \neq q$ and $n = p*q$, then $\Phi(n) = (p-1)(q-1)$
Example: $\Phi(15) = ?$

Here,

$15 = 3 * 5$

p = 3 and q = 5 then

$\Phi(15) = (3-1)(5-1)$

$= 2 * 4$

$= 8$

Programming Language: C

IDE: Dev C++

Code :

```c
#include <stdio.h>

// gcd calculation
int gcd(int a, int b)
{
   if (a == 0)
      return b;
   return gcd(b % a, a);
}

int phi(unsigned int n)
{
   int i;
   unsigned int result = 1;
   for (i = 2; i < n; i++)
      if (gcd(i, n) == 1)
         result++;
   return result;
}
int main()
{
   int n;
   printf("Enter the number:");
   scanf("%d", &n);
   printf("phi(%d) = %d\n", n, phi(n));
   return 0;
}
```

Output

```
C:\Users\Aayush\Desktop\MB

Enter the number:45
phi(45) = 24

--------------------------------
Process exited after 2.272 seconds with return value 0
Press any key to continue . . .
```

Madan Bhandari Memorial College

Name: Aayush Basnet **Date:**

Lab 9: WAP for primitive root.

Theory:

Given a prime number n, the task is to find its primitive root under modulo n. The primitive root of a prime number n is an integer r between [1, n-1] such that the values of r^x (mod n) where x is in the range [0, n-2] are different. -1 if n is a non-prime number.

Example:

n = 7

r = 3 then,
30 (mod 7) = 1

31 (mod 7) = 3

32 (mod 7) = 2

33 (mod 7) = 6

34 (mod 7) = 4

35 (mod 7) = 5

Programming Language: C

IDE: Dev C++

Code :

```
#include <stdio.h>
#include <stdbool.h>
#include <math.h>
#include <string.h>
#include <stdlib.h>

bool isPrime(int n) {
   int i;
   if (n <= 1) return false;
   if (n <= 3) return true;

   if (n%2 == 0 || n%3 == 0) return false;

   for (i = 5; i <= sqrt(n); i = i + 6)
      if (n%i == 0 || n%(i+2) == 0)
         return false;

   return true;
}

int power(int x, unsigned int y, int p) {
   int res = 1;
   x = x % p;

   while (y > 0) {
      if (y & 1)
         res = (res*x) % p;

      y = y >> 1;
      x = (x*x) % p;
   }
   return res;
}
void findPrimefactors(int* s, int* size, int n) {
   int i;

   while (n % 2 == 0) {
      s[(*size)++] = 2;
  n = n / 2;
   }
```

```
    for (i = 3; i <= sqrt(n); i = i + 2) {
        while (n % i == 0) {
            s[(*size)++] = i;
            n = n / i;
        }
    }

    if (n > 2)
        s[(*size)++] = n;
}

int findPrimitive(int n) {
    int r, i;
    int s[20];
    int size = 0;

    if (!isPrime(n)) return -1;

    int phi = n - 1;
    findPrimefactors(s, &size, phi);

    for (r = 1; r <= phi; r++) {
        bool flag = true;
        for (i = 0; i < size; i++) {
            if (power(r, phi / s[i], n) == 1) {
                flag = false;
                break;
            }
        }
        if (flag == true)
            return r;
    }
    return -1;
}

int main() {
    int n;
    printf("Enter the number: ");

scanf("%d", &n);
    printf("Smallest primitive root of %d is %d\n", n, findPrimitive(n));

    return 0;

}
```

Output

```
C:\Users\Aayush\Desktop\MB

Enter the number:25
 Smallest primitive root of 25 is -1

--------------------------------
Process exited after 3.751 seconds with return value 0
Press any key to continue . . .
```

```
C:\Users\Aayush\Desktop\MB

Enter the number:5
 Smallest primitive root of 5 is 2

--------------------------------
Process exited after 0.7362 seconds with return value 0
Press any key to continue . . .
```
