**Assignment 8**

1. **Miller–Rabin primality test**

**Code:**

#include <bits/stdc++.h>

using namespace std;

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;

}

bool miillerTest(int d, int n)

{

int a = 2 + rand() % (n - 4);

int x = power(a, d, n);

if (x == 1 || x == n-1)

return true;

while (d != n-1)

{

x = (x \* x) % n;

d \*= 2;

if (x == 1) return false;

if (x == n-1) return true;

}

return false;

}

bool isPrime(int n, int k)

{

if (n <= 1 || n == 4) return false;

if (n <= 3) return true;

int d = n - 1;

while (d % 2 == 0)

d /= 2;

for (int i = 0; i < k; i++)

if (!miillerTest(d, n))

return false;

return true;

}

int main()

{

int k;

cout<<"Enter the number of iterations : ";

cin>>k;

int t=1;

while(t)

{

cout<<"\nEnter the number : ";

int n;

cin>>n;

if(isPrime(n,k))

cout<<"The number is prime\n";

else

cout<<"The number is not prime\n";

int x;

cout<<"press 1 for continue, 0 to exit: ";

cin>>x;

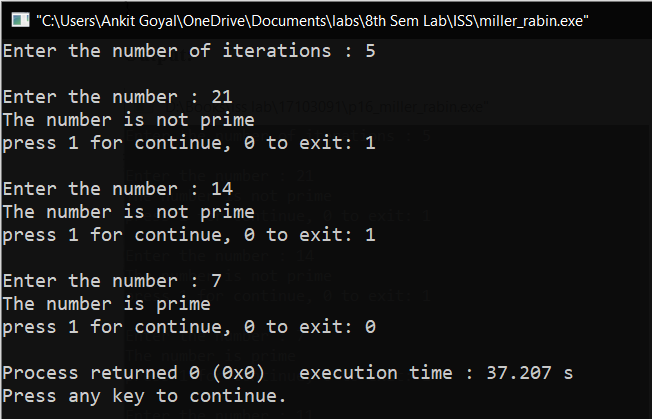
t=x;

}

return 0;

}

**Output:**



1. **Chinese Remainder Theorem**

**Code:**

#include <bits/stdc++.h>

using namespace std;

int inv(int a, int m)

{

int m0 = m, t, q;

int x0 = 0, x1 = 1;

if (m == 1)

return 0;

while (a > 1) {

q = a / m;

t = m;

m = a % m, a = t;

t = x0;

x0 = x1 - q \* x0;

x1 = t;

}

if (x1 < 0)

x1 += m0;

return x1;

}

int findMinX(int num[], int rem[], int k)

{

int prod = 1;

for (int i = 0; i < k; i++)

prod \*= num[i];

int result = 0;

for (int i = 0; i < k; i++) {

int pp = prod / num[i];

result += rem[i] \* inv(pp, num[i]) \* pp;

}

return result % prod;

}

int main(void)

{

int n;

cout<<"Enter the number of equations : ";

cin>>n;

cout<<"\nEnter the numbers and their remainders in each equation:\n ";

int num[n],rem[n];

for(int i=0;i<n; ++i)

{

cin>>num[i]>>rem[i];

}

cout << "x is " << findMinX(num, rem, n);

return 0;

}

**Output:**

