Problem 1.

Solution :-

using System.Numerics;

class Program

{

static void Main()

{

HashSet<string> distinctTerms = new HashSet<string>();

for (int a = 2; a <= 100; a++)

{

for (int b = 2; b <= 100; b++)

{

distinctTerms.Add(BigInteger.Pow(a, b).ToString());

}

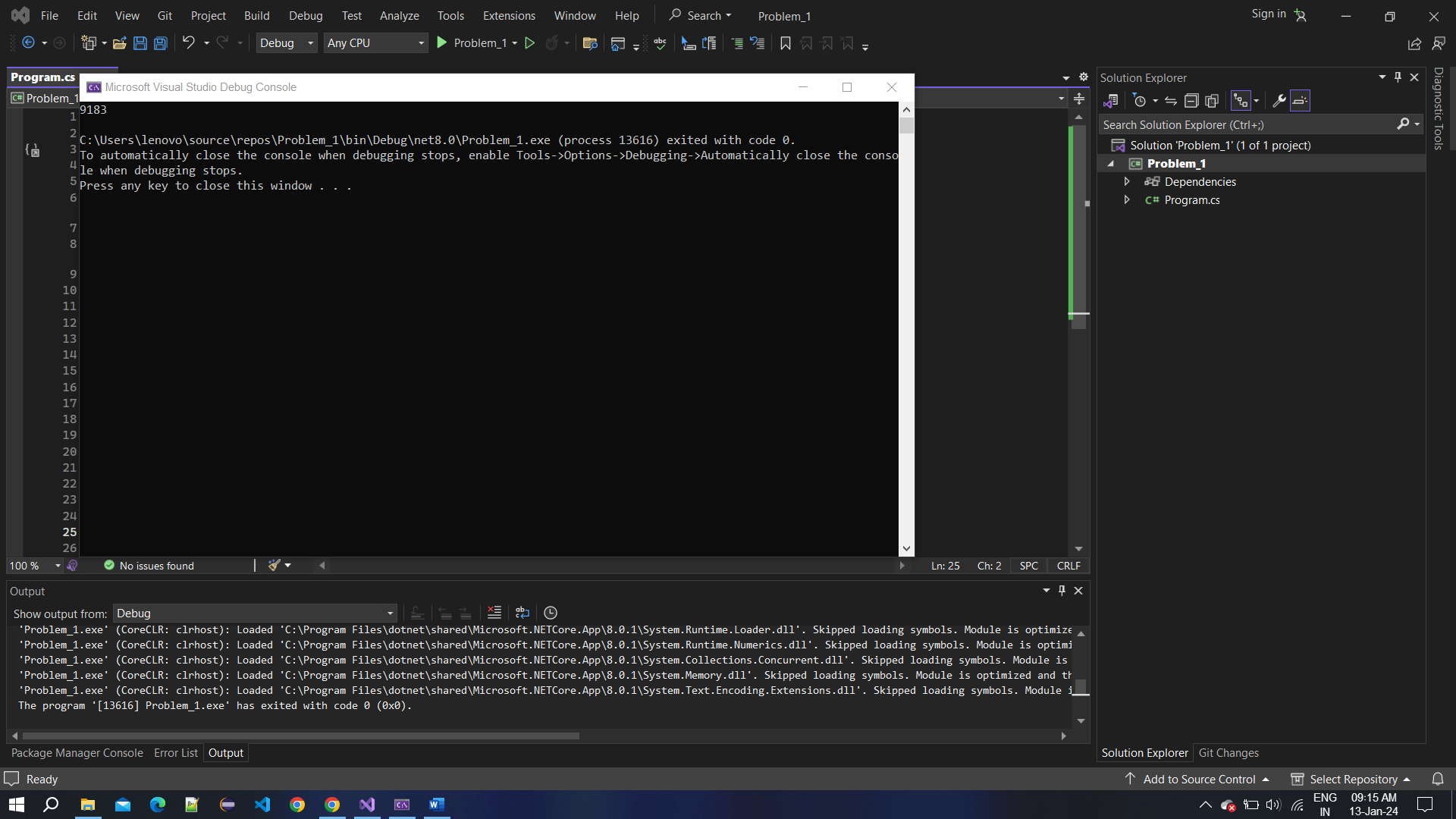
}

int result = distinctTerms.Count;

Console.WriteLine(result);

}

}



Problem 2

Solution :-

using System;

class Program

{

static void Main()

{

int result = SumOfDigitFactorials();

Console.WriteLine(result);

}

static int SumOfDigitFactorials()

{

int sum = 0;

// Factorials for digits 0-9

int[] factorials = new int[10];

for (int i = 0; i < 10; i++)

{

factorials[i] = Factorial(i);

}

// Iterate through numbers to find the sum

for (int num = 10; num < 100000; num++)

{

if (IsSumOfDigitFactorials(num, factorials))

{

sum += num;

}

}

return sum;

}

static int Factorial(int n)

{

if (n == 0 || n == 1)

{

return 1;

}

else

{

return n \* Factorial(n - 1);

}

}

static bool IsSumOfDigitFactorials(int num, int[] factorials)

{

int sum = 0;

int originalNum = num;

while (num > 0)

{

int digit = num % 10;

sum += factorials[digit];

num /= 10;

}

return sum == originalNum;

}

}

