Даны различные натуральные числа А, В, N: 1 < А, B < 10, 10 < N < 100.

Сумма нескольких чисел равна N, каждое из чисел равно А или B.

Найти наибольшее возможное значение произведения этих чисел. Если такая

ситуация невозможна, то вывести 0.

**Input**

Три целых числа в одной строке -- А, В, N.

**Output**

Вывести целое число -- ответ задачи.

**Sample input I**

3, 4 ,12

**Sample output I**

81

**Sample input II**

3, 5, 13

**Sample output II**

75

Решение С#:

using System;

using System.Collections.Generic;

using System.Linq;

namespace GreatestMultiplication

{

/// <summary>

/// Даны различные натуральные числа А, В, N: 1 < А, B < 10, 10 < N < 100.

/// Сумма нескольких чисел равна N, каждое из чисел равно А или B.

/// Найти наибольшее возможное значение произведения этих чисел.Если такая

/// ситуация невозможна, то вывести 0.

/// </summary>

class Program

{

static void Main(string[] args)

{

while (true)

{

int A = 0;

while (true)

{

Console.Write("Задайте A: ");

A = int.Parse(Console.ReadLine());

if (A <= 1)

Console.WriteLine("A должно быть больше 1! Попробуйте еще раз...");

else

break;

}

int B = 0;

while (true)

{

Console.Write("Задайте B: ");

B = int.Parse(Console.ReadLine());

if (B >= 10)

Console.WriteLine("B должно быть меньше 10! Попробуйте еще раз...");

else

break;

}

int N = 0;

while (true)

{

Console.Write("Задайте N: ");

N = int.Parse(Console.ReadLine());

if (!(N > 10 && N < 100))

Console.WriteLine("N должно быть на интервале (10; 100)! Попробуйте еще раз...");

else

break;

}

List<int> values = new List<int>();//список значений, участвоваших в сумме

//добавим в массив values дополнительные натуральные числа A или B для дальнейшего нахождения сумм

//для этого нужно понять: сколько возможно максимально добавить таких чисел

int limA = N / A;//элементов А

int limB = N / B;//элементов B

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

{

values.Add(A);

}

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

{

values.Add(B);

}

List<int> multis = new List<int>();//список произведений

List<int> sums = new List<int>();//список сумм

for (int i = 0; i < values.Count; ++i)

{

int res = Sum(sums, multis, values, i, 0, 1, N);

}

if (multis.Count != 0)

Console.WriteLine("\nОтвет: " + multis.Max());

else

Console.WriteLine("\nОтвет: 0");

Console.Write("\n");

}

Console.ReadKey();

}

/// <summary>

/// Рекурсивная функция поиска всех возможных сумм и произведений

/// </summary>

/// <param name="sums">Список сумм</param>

/// <param name="multis">Список произведений</param>

/// <param name="values">Список элементов</param>

/// <param name="index">Индекс</param>

/// <param name="curSum">Текущая сумма</param>

/// <param name="curMulti">Текущее произведение</param>

/// <param name="N">Искомая сумма</param>

/// <returns></returns>

static int Sum(List<int> sums, List<int> multis, List<int> values, int index, int curSum, int curMulti, int N)

{

if (index < values.Count)

{

if (values.Count > 10)

{

Console.Write("\x000DПодождите... Идет поиск. Кол-во найденных вариантов: " + multis.Count);

}

curSum += values[index];

curMulti \*= values[index];

for (int i = index + 1; i <= values.Count; ++i)

{

int res = Sum(sums, multis, values, i, curSum, curMulti, N);

if (res == N)

{

sums.Add(res);//добавляем сумму с список

multis.Add(curMulti);//добавляем произведение в список

}

}

}

return curSum;

}

}

}

Результат:

