Министерство науки и высшего образования Российской Федерации

Федеральное государственное образовательное учреждение

высшего образования

**ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

**Дисциплина: Алгоритмы и структуры данных**

**Отчёт по лабораторной работе №7**

**Тема: «Рекурсия»**

**Вариант 2**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил студент  группы ИТб 2302-02-20 |  | Ердяков Роман Александрович |  |
|  |  |
| Проверила |  | Кашина Елена Вячеславовна |  |

СОДЕРЖАНИЕ

[1 Задание 3](#_Toc200883332)

[2 Тестирование (задача 1) 5](#_Toc200883333)

[3 Программа на C# (задача 1): 6](#_Toc200883334)

[4 Тестирование (задача 2) 7](#_Toc200883335)

[5 Программа на C# (задача 2): 8](#_Toc200883336)

[6 Тестирование (задача 4) 9](#_Toc200883337)

[7 Программа на C# (задача 4): 10](#_Toc200883338)

[8 Тестирование (задача 6) 11](#_Toc200883339)

[9 Программа на C# (задача 6): 12](#_Toc200883340)

[10 Тестирование (задача 8) 13](#_Toc200883341)

[11 Программа на C# (задача 8): 14](#_Toc200883342)

[12 Тестирование (задача 10) 15](#_Toc200883343)

[13 Программа на C# (задача 10): 16](#_Toc200883344)

[14 Тестирование (задача 12) 17](#_Toc200883345)

[15 Программа на C# (задача 12): 18](#_Toc200883346)

[16 Тестирование (задача 14) 19](#_Toc200883347)

[17 Программа на C# (задача 14): 20](#_Toc200883348)

[18 Тестирование (задача 16) 21](#_Toc200883349)

[19 Программа на C# (задача 16): 22](#_Toc200883350)

[20 Тестирование (задача 18) 23](#_Toc200883351)

[21 Программа на C# (задача 18): 24](#_Toc200883352)

[22 Вывод 25](#_Toc200883353)

# Задание

Задача 1.

Имеется 10 белых и 5 черных шаров. Сколькими способами можно выбрать 7 шаров, чтобы среди них были 3 черных?

Задача 2.

Написать рекурсивную программу, для вычисления аn (возведение целого числа а в целую неотрицательную степень n)

Задача 4.

Составить программу перевода числа из 10-ой системы в n-ичную, где 2<=n<10

Задача 6.

Написать рекурсивную программу вычисления суммы n первых членов арифметической прогрессии.

Задача 8.

Написать рекурсивную программу поиска минимального элемента массива.

Задача 10.

Проверьте, являются ли два данных числа взаимно простыми.

Задача 12.

Написать вариант алгоритма Евклида, основанный на соотношениях: НОД(2\*а, 2\*b)= 2\*НОД(a,b); НОД(2\*a,b) = НОД(a,b), при нечётном b, не включающий деления с остатком, а использующий лишь деление на 2 и проверку чётности.

Задача 14.

Напишите рекурсивную программу вычисления n-го члена геометрической прогрессии (bn= b1\*q^(n-1)), суммы её n первых членов (рекурсивно) и суммы ее членов, начиная с i-гo по k-ый.

Задача 16.

Описать рекурсивную функцию поиска индекса минимального элемента массива.

Задача 18.

Написать функцию, которая находит НОД двух неотрицательных чисел по алгоритму Евклида (второй способ нахождения НОД – делением): Пусть х и у одновременно неравные нулю целые неотрицательные числа и пусть х>=y, тогда если у=0, то НОД(х,у)=х, а если у<>0, то для чисел х, у и r, где r – остаток от деления х на у выполняется равенство: НОД(х,у)= НОД(у,r). (Другими словами: Последний ненулевой остаток, деление на который осуществляется нацело и является НОД чисел х и у)

# Тестирование (задача 1)

Примеры выполнения программы представлены на рисунке 1.



Рисунок 1 – Экранная форма программы

# Программа на C# (задача 1):

class Program

{

static void Main()

{

int totalWhite = 10;

int totalBlack = 5;

int selectTotal = 7;

int selectBlack = 3;

int selectWhite = selectTotal - selectBlack;

int blackWays = Combination(totalBlack, selectBlack);

int whiteWays = Combination(totalWhite, selectWhite);

int totalWays = blackWays \* whiteWays;

Console.WriteLine($"Число способов выбрать {selectTotal} шаров с {selectBlack} черными: {totalWays}");

}

static int Combination(int n, int k)

{

if (k == 0 || k == n)

return 1;

if (k > n)

return 0;

return Combination(n - 1, k - 1) + Combination(n - 1, k);

}

}

# Тестирование (задача 2)

Примеры выполнения программы представлены на рисунках 2, 3.



Рисунок 3 – Экранная форма программы



Рисунок 4 – Экранная форма программы

# Программа на C# (задача 2):

using System;

class Program

{

static void Main()

{

Console.Write("Введите целое число a (основание): ");

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

Console.Write("Введите целое неотрицательное число n (степень): ");

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

if (n < 0)

{

Console.WriteLine("Степень должна быть неотрицательной.");

return;

}

int result = Power(a, n);

Console.WriteLine($"{a}^{n} = {result}");

}

static int Power(int a, int n)

{

if (n == 0)

return 1;

return a \* Power(a, n - 1);

}

}

# Тестирование (задача 4)

Примеры выполнения программы представлены на рисунках 4, 5.



Рисунок 4 – Экранная форма программы



Рисунок 5 – Экранная форма программы

# Программа на C# (задача 4):

class Program

{

static void Main()

{

Console.Write("Введите десятичное число: ");

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

Console.Write("Введите основание системы (2-9): ");

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

if (baseN < 2 || baseN > 9)

{

Console.WriteLine("Основание должно быть от 2 до 9.");

return;

}

string result = ConvertToBaseN(number, baseN);

Console.WriteLine($"Число {number} в системе счисления с основанием {baseN}: {result}");

}

static string ConvertToBaseN(int number, int baseN)

{

if (number < baseN)

return number.ToString();

else

return ConvertToBaseN(number / baseN, baseN) + (number % baseN).ToString();

}

}

# Тестирование (задача 6)

Примеры выполнения программы представлены на рисунках 6, 7.

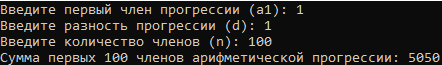


Рисунок 6 – Экранная форма программы

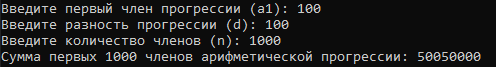


Рисунок 7 – Экранная форма программы

# Программа на C# (задача 6):

class Program

{

static void Main()

{

Console.Write("Введите первый член прогрессии (a1): ");

int a1 = int.Parse(Console.ReadLine());

Console.Write("Введите разность прогрессии (d): ");

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

Console.Write("Введите количество членов (n): ");

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

int sum = SumArithmeticProgression(a1, d, n);

Console.WriteLine($"Сумма первых {n} членов арифметической прогрессии: {sum}");

}

static int SumArithmeticProgression(int a1, int d, int n)

{

if (n == 1)

return a1;

else

return a1 + (n - 1) \* d + SumArithmeticProgression(a1, d, n - 1);

}

}

# Тестирование (задача 8)

Примеры выполнения программы представлены на рисунках 8 и 9.



Рисунок 8 – Экранная форма программы



Рисунок 9 – Экранная форма программы

# Программа на C# (задача 8):

class Program

{

static void Main()

{

Random rnd = new Random();

int size = 10;

int[] arr = new int[size];

Console.WriteLine("Сгенерированный массив:");

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

{

arr[i] = rnd.Next(1, 100);

Console.Write(arr[i] + " ");

}

Console.WriteLine();

int min = FindMin(arr, 0, arr[0]);

Console.WriteLine($"Минимальный элемент массива: {min}");

}

static int FindMin(int[] arr, int index, int currentMin)

{

if (index == arr.Length)

return currentMin;

if (arr[index] < currentMin)

currentMin = arr[index];

return FindMin(arr, index + 1, currentMin);

}

}

# Тестирование (задача 10)

Примеры выполнения программы представлены на рисунках 10, 11.



Рисунок 10 – Экранная форма программы



Рисунок 11 – Экранная форма программы

# Программа на C# (задача 10):

class Program

{

static void Main()

{

Console.Write("Введите первое число: ");

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

Console.Write("Введите второе число: ");

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

int gcd = GCD(Math.Abs(a), Math.Abs(b));

if (gcd == 1)

Console.WriteLine("Числа взаимно простые.");

else

Console.WriteLine("Числа НЕ взаимно простые.");

}

static int GCD(int a, int b)

{

if (b == 0)

return a;

return GCD(b, a % b);

}

}

# Тестирование (задача 12)

Примеры выполнения программы представлены на рисунках 12, 13.



Рисунок 12 – Экранная форма программы



Рисунок 13 – Экранная форма программы

# Программа на C# (задача 12):

class Program

{

static void Main()

{

Console.Write("Введите число a: ");

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

Console.Write("Введите число b: ");

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

int result = GCD(a, b);

Console.WriteLine($"НОД({a}, {b}) = {result}");

}

static int GCD(int a, int b)

{

a = Math.Abs(a);

b = Math.Abs(b);

if (a == 0) return b;

if (b == 0) return a;

if (IsEven(a) && IsEven(b))

return 2 \* GCD(a / 2, b / 2);

if (IsEven(a) && !IsEven(b))

return GCD(a / 2, b);

if (!IsEven(a) && IsEven(b))

return GCD(a, b / 2);

if (a > b)

return GCD((a - b) / 2, b);

else

return GCD((b - a) / 2, a);

}

static bool IsEven(int x)

{

return (x & 1) == 0;

}

}

# Тестирование (задача 14)

Примеры выполнения программы представлены на рисунках 14, 15.

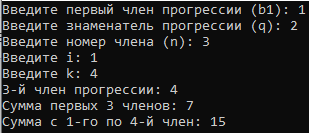


Рисунок 14 – Экранная форма программы

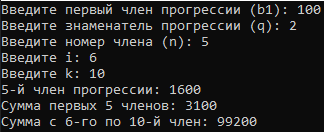


Рисунок 15 – Экранная форма программы

# Программа на C# (задача 14):

class Program

{

static void Main()

{

Console.Write("Введите первый член прогрессии (b1): ");

double b1 = double.Parse(Console.ReadLine());

Console.Write("Введите знаменатель прогрессии (q): ");

double q = double.Parse(Console.ReadLine());

Console.Write("Введите номер члена (n): ");

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

Console.Write("Введите i: ");

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

Console.Write("Введите k: ");

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

double nth = GetMember(b1, q, n);

Console.WriteLine($"{n}-й член прогрессии: {nth}");

double sumN = Sum(b1, q, n);

Console.WriteLine($"Сумма первых {n} членов: {sumN}");

double sumFromIToK = Sum(b1, q, k) - Sum(b1, q, i - 1);

Console.WriteLine($"Сумма с {i}-го по {k}-й член: {sumFromIToK}");

}

static double GetMember(double b1, double q, int n)

{

if (n == 1) return b1;

return q \* GetMember(b1, q, n - 1);

}

static double Sum(double b1, double q, int n)

{

if (n == 0) return 0;

return GetMember(b1, q, n) + Sum(b1, q, n - 1);

}

}

# Тестирование (задача 16)

Примеры выполнения программы представлены на рисунках 16, 17.



Рисунок 16 – Экранная форма программы



Рисунок 17 – Экранная форма программы

# Программа на C# (задача 16):

class Program

{

static void Main()

{

Random rnd = new Random();

int[] array = new int[10];

for (int i = 0; i < array.Length; i++)

{

array[i] = rnd.Next(1, 100);

}

Console.WriteLine("Массив: " + string.Join(", ", array));

int minIndex = FindMinIndex(array, 0, array.Length - 1);

Console.WriteLine($"Минимальный элемент: {array[minIndex]} на индексе {minIndex}");

}

static int FindMinIndex(int[] array, int start, int end)

{

if (start == end)

return start;

int minRestIndex = FindMinIndex(array, start + 1, end);

return array[start] < array[minRestIndex] ? start : minRestIndex;

}

}

# Тестирование (задача 18)

Примеры выполнения программы представлены на рисунках 18, 19.



Рисунок 18 – Экранная форма программы, стартовый экран



Рисунок 19 – Экранная форма программы, нажали кнопку запуска

# Программа на C# (задача 18):

class Program

{

static void Main()

{

Console.Write("Введите первое число (x): ");

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

Console.Write("Введите второе число (y): ");

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

if (x == 0 && y == 0)

{

Console.WriteLine("НОД не определён для двух нулей.");

return;

}

int result = GCD(x, y);

Console.WriteLine($"НОД({x}, {y}) = {result}");

}

static int GCD(int x, int y)

{

if (y == 0)

return x;

return GCD(y, x % y);

}

}

# Вывод

В ходе выполнения этих задач научились применять рекурсию для решения различных типов задач. Научились с помощь рекурсии возведение в степень, находить суммы арифметической и геометрической прогрессий, переводить чисела в другую систему счисления. Так же научились нахождению НОД по разным методам, определение взаимной простоты чисел, поиск минимума и его индекса в массиве.