Министерство образования ХХХХ

ГБПОУ ХХХХ «ХХХХ»

09.02.07

Дисциплина: Основы алгоритмизации и программирования

ЛАБОРАТОРНЫЕ РАБОТЫ

ОТЧЁТ

ККОО.ОАХХХХ.000

|  |  |
| --- | --- |
| Студент: |  |
| Преподаватель: |  |
| Дата: | Оценка: |

2021

Содержание

|  |  |  |
| --- | --- | --- |
| 1 | Лабораторная работа 1 | 3 |
| 2 | Лабораторная работа 2 | 8 |
| 3 | Лабораторная работа 3 | 28 |
| 4 | Лабораторная работа 4 | 34 |
| 5 | Лабораторная работа 5-6 | 55 |
| 6  7 | Лабараторная работа 7  Лабараторная работа 8 | 94  151 |
| 8 | Лабараторная работа 9 | 158 |

Лабораторная работа №1

Тема: Разработка простейших консольных приложений.

Цель: Овладение возможностью использования команд ввода-вывода.

Часть 2.

1. Написать программу, которая запрашивает с клавиатуры два целых числа и выводит на экран сумму данных чисел:

string first;

string second;

Console.WriteLine("Введите первую переменную: ");

first = Console.ReadLine();

int first\_int = Convert.ToInt32(first);

Console.WriteLine("Введите вторую переменную: ");

second = Console.ReadLine();

int second\_int = Convert.ToInt32(second);

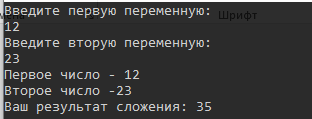
int result = first\_int + second\_int;

Console.WriteLine($"Первое число - {first\_int}");

Console.WriteLine($"Второе число -{second\_int}");

Console.WriteLine($"Ваш результат сложения: {result}");

Console.ReadKey();



1. Написать программу, которая запрашивает с клавиатуры три целых числа и выводит на экран сумму данных чисел:

string first;

string second;

string third;

Console.WriteLine("Введите первую переменную: ");

first = Console.ReadLine();

int first\_int = Convert.ToInt32(first);

Console.WriteLine("Введите вторую переменную: ");

second = Console.ReadLine();

int second\_int = Convert.ToInt32(second);

Console.WriteLine("Введите третью переменную: ");

third = Console.ReadLine();

int third\_int = Convert.ToInt32(third);

int result = first\_int + second\_int + third\_int;

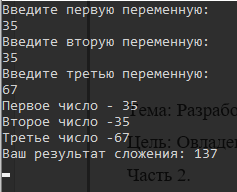
Console.WriteLine($"Первое число - {first\_int}");

Console.WriteLine($"Второе число -{second\_int}");

Console.WriteLine($"Третье число -{third\_int}");

Console.WriteLine($"Ваш результат сложения: {result}");

Console.ReadKey();



1. Написать программу, которая запрашивает с клавиатуры два вещественных числа, и выводит на экран результат деления первого числа на второе (вещественные числа выводятся с точностью до 3 знаков после запятой):

string first;

string second;

Console.WriteLine("Введите первую переменную: ");

first = Console.ReadLine();

double first\_double = Convert.ToDouble(first);

Console.WriteLine("Введите вторую переменную: ");

second = Console.ReadLine();

double second\_double = Convert.ToDouble(second);

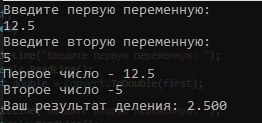
double result = first\_double / second\_double;

Console.WriteLine($"Первое число - {first\_double:f3}");

Console.WriteLine($"Второе число -{second\_double:f3}");

Console.WriteLine($"Ваш результат деления: {result:f3}");

Console.ReadKey();



1. Написать программу, которая запрашивает с клавиатуры три вещественных числа и выводит на экран следующее сообщение (вещественные числа выводятся с точностью до 2 знаков после запятой):

string first;

string second;

string third;

Console.WriteLine("Введите первую переменную: ");

first = Console.ReadLine();

Double first\_f = Convert.ToDouble(first);

Console.WriteLine("Введите вторую переменную: ");

second = Console.ReadLine();

Double second\_f = Convert.ToDouble(second);

Console.WriteLine("Введите третью переменную: ");

third = Console.ReadLine();

Double third\_f = Convert.ToDouble(third);

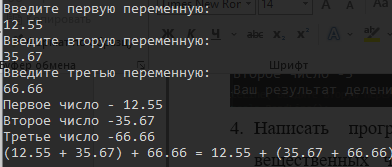
Console.WriteLine($"Первое число - {first\_f}");

Console.WriteLine($"Второе число -{second\_f}");

Console.WriteLine($"Третье число -{third\_f}");

Console.WriteLine($"({first\_f:f2} + {second\_f:f2}) + {third\_f:f2} = {first\_f:f2} + ({second\_f:f2} + {third\_f:f2})");

Console.ReadKey();



1. Написать программу, которая запрашивает с клавиатуры номинал купюры и количество купюр, и выводит экран следующее сообщение:

Console.WriteLine("Введите номинал купюры:");

Double currency = Convert.ToDouble(Console.ReadLine());

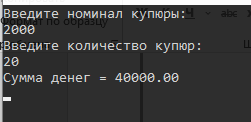
Console.WriteLine("Введите количество купюр:");

int count = Convert.ToInt32(Console.ReadLine());

Double result = count \* currency;

Console.WriteLine($"Сумма денег = {result:f2}");

Console.ReadKey();



1. Написать программу, которая запрашивает с клавиатуры сумму вклада и процент по вкладу, и выводит на экран начисленную в конце года сумму:

Console.WriteLine("Введите сумму вклада:");

double sum = Convert.ToDouble(Console.ReadLine());

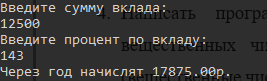
Console.WriteLine("Введите процент по вкладу:");

double percent = Convert.ToDouble(Console.ReadLine());

double result = sum \* (percent / 100);

Console.WriteLine($"Через год начислят {result:f2}" + "p.");

Console.ReadKey();



1. Написать программу, которая запрашивает с клавиатуры имя человека и его возраст, и выводит на экран следующее сообщение (в примере текущим годом считается 2009):

Console.Write("Как тебя зовут?");

string name = Console.ReadLine();

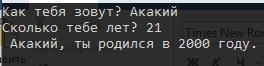
Console.Write("Сколько тебе лет?");

int age = Convert.ToInt32(Console.ReadLine());

int dob = 2021 - age;

Console.WriteLine($"{name}, ты родился в {dob} году.");

Console.ReadKey();

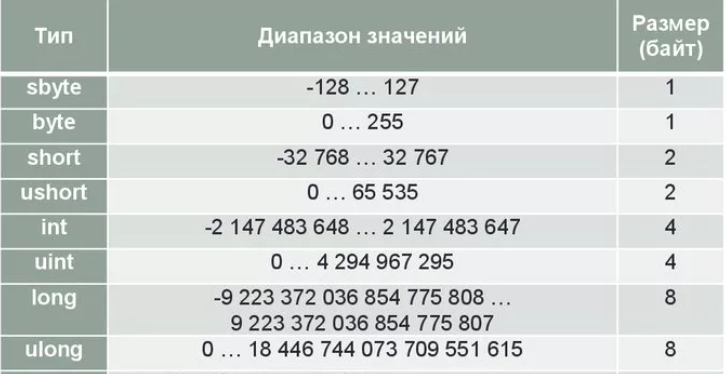


Лабораторная работа №2.

Часть 2.1

Тема: Целый тип данных

Существует восемь целых типов.Они отличаются диапазоном значений, а значит, и размером памяти, отводимой для их представления.



Операции с величинами целого типа: сложение (+), вычитание (-), умножение (\*), нахождение целой части деления (/)*,* нахождение остатка от деления (%)*.*

В следующей программе определяются цифры трехзначного числа. Можно ее использовать и для определения цифр двузначного числа, просто цифра сотен в этом случае равна нулю, и это делается проще.

int a, one, dec, hun, rez;

Console.WriteLine("Введите трехзначное число");

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

one = a % 10;

Console.WriteLine("Цифра единиц числа - {0}", one);

dec = (a / 10) % 10;

Console.WriteLine("Цифра десятков числа - {0}", dec);

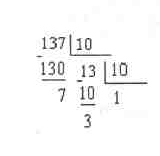
hun = a / 100;

Console.WriteLine("Цифра сотен числа - {0}", hun);

rez = hun \* 100 + dec \* 10 + one;

Console.WriteLine("это тоже число - {0}", rez);

Console.ReadKey();



Например, если Вы введете число 137, то значение переменной oneбудет равно 7, dec *—* 3 и hun *—* 1. Вспомните деление чисел столбиком.

Экспериментальный раздел работы:

1. Измените программудля нахождения цифр двузначного числа. Сохраните ее*.*

int a, one, dec, hun, rez;

Console.WriteLine("Введите двузначное число");

a = Convert.ToInt32(Console.ReadLine());

one = a % 10;

Console.WriteLine("Цифра единиц числа - {0}", one);

dec = (a / 10) % 10;

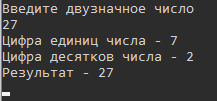
Console.WriteLine("Цифра десятков числа - {0}", dec);

hun = a / 100;

rez = dec \* 10 + one;

Console.WriteLine("Результат - {0}", rez);

Console.ReadKey();



1. Измените программу для нахождения цифр четырехзначного числа. Сохраните ее*.*

int a, one, dec, hun, tis, rez;

Console.WriteLine("Введите четырехзначное число");

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

one = a % 10;

Console.WriteLine("Цифра единиц числа - {0}", one);

dec = (a / 10) % 10;

Console.WriteLine("Цифра десятков числа - {0}", dec);

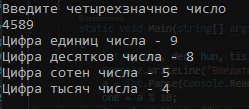
hun = (a / 100) % 10;

Console.WriteLine("Цифра сотен числа - {0}", hun);

tis = a / 1000;

Console.WriteLine("Цифра тысяч числа - {0}", tis);

Console.ReadKey();



Задания для самостоятельной работы

1. Дано двузначное число. Определить:

* сумму и произведение цифр числа;
* число, образованное перестановкой цифр исходного числа.

int chislo, a, b, result;

Console.WriteLine("Введите двузначное число:");

chislo = Convert.ToInt32(Console.ReadLine());

a = chislo % 10;

b = chislo / 10;

Console.WriteLine($"Ваше число - {chislo}\nЕдиницы - {a} \nДесятки - {b}");

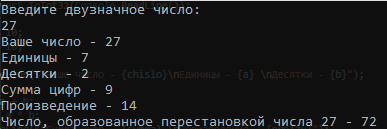
int summa = a + b;

int proizv = a \* b;

Console.WriteLine($"Сумма цифр - {summa} \nПроизведение - {proizv}");

Console.WriteLine($"Число, образованное перестановкой числа {chislo} - {a}{b}");

Console.ReadKey();



2. Дано трехзначное число. Определить:

* сумму и произведение цифр числа;
* число, образованное перестановкой цифр исходного числа;
* число, полученное перестановкой цифр десятков и единиц;
* число, полученное перестановкой цифр сотен и десятков;
* четырехзначное число, полученное приписыванием цифры единиц в качестве цифры тысяч (например, из числа 137 необходимо получить число 7137).

Примечание

## Сколько различных чисел можно получить из трехзначного числа

путем перестановки цифр? Ответ: 6

int chislo, a, b, c;

Console.WriteLine("Введите трехзначное число:");

chislo = Convert.ToInt32(Console.ReadLine());

a = chislo % 10;

b = chislo / 10 % 10;

c = chislo / 100;

Console.WriteLine($"Ваше число - {chislo}\nЕдиницы - {a} \nДесятки - {b} \nСотни - {c} ");

int summa = a + b + c;

int proizv = a \* b \* c;

Console.WriteLine($"Сумма цифр - {summa} \nПроизведение - {proizv}");

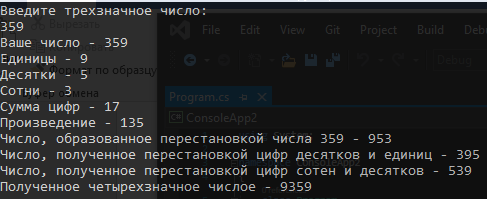
Console.WriteLine($"Число, образованное перестановкой числа {chislo} - {a}{b}{c}");

Console.WriteLine($"Число, полученное перестановкой цифр десятков и единиц - {c}{a}{b}");

Console.WriteLine($"Число, полученное перестановкой цифр сотен и десятков - {b}{c}{a}");

Console.WriteLine($"Полученное четырехзначное числое - {a}{c}{b}{a}");

Console.ReadKey();



1. Решить задачу 2 (кроме последнего пункта) для четырехзначных чисел.

int chislo, a, b, c, d;

Console.WriteLine("Введите четырехзначное число:");

chislo = Convert.ToInt32(Console.ReadLine());

a = chislo % 10;

b = (chislo / 10) % 10;

c = (chislo / 100) % 10;

d = chislo / 1000;

Console.WriteLine($"Ваше число - {chislo}\nЕдиницы - {a} \nДесятки - {b} \nСотни - {c} \nТысячи - {d} ");

int summa = a + b + c + d;

int proizv = a \* b \* c \* d;

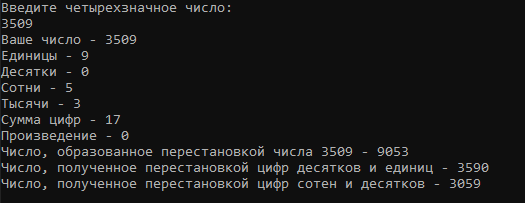
Console.WriteLine($"Сумма цифр - {summa} \nПроизведение - {proizv}");

Console.WriteLine($"Число, образованное перестановкой числа {chislo} - {a}{b}{c}{d}");

Console.WriteLine($"Число, полученное перестановкой цифр десятков и единиц - {d}{c}{a}{b}");

Console.WriteLine($"Число, полученное перестановкой цифр сотен и десятков - {d}{b}{c}{a}");

Console.ReadKey();



1. Арифметическая прогрессия — это последовательность чисел, в которой разность между последующим и предыдущим элементами остается неизменной. Последовательность 12, 15, 18, 21, 24, ... является арифметической прогрессией, 12 — первый член прогрессии (a1)*,* разность прогрессии равна 3. Любой член прогрессии вычисляется по формуле an=a1+d\*(n-l), где d — разность прогрессии, n — номер взятого члена. Даны a1 и d. Найти n, при котором значение аn выходит за диапазон типа short (экспериментальным путем).

int a\_1 = 12;

int d = 3;

Console.WriteLine("Введите номер взятого члена:");

int n = Convert.ToInt32(Console.ReadLine());

short a\_n = (short)(a\_1 + d \* (n - 1));

Console.WriteLine($"Арифметическая прогрессия = {a\_n}");

//10920





1. Сумма первых n членов арифметической прогрессии вычисляется по формуле Sn=(a1+an)\*n/2. Даны а1 и d. Найти n, при котором значение Sn выходит за диапазон типа short (экспериментальным путем).

int a\_1 = 12;

int d = 3;

Console.WriteLine("Введите номер взятого члена:");

int n = Convert.ToInt32(Console.ReadLine());

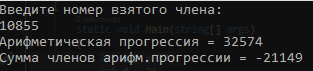
short a\_n = (short)(a\_1 + d \* (n - 1));

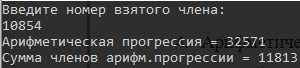
Console.WriteLine($"Арифметическая прогрессия = {a\_n}");

short s\_n = (short)((a\_1 + a\_n) \* n / 2);

Console.WriteLine($"Сумма членов арифм.прогрессии = {s\_n}");

//10855





Часть 2.2

Разработать консольное приложение, с помощью которого можно выполнить математические вычисления (см. вариант).

При выполнении заданий допустимо использовать справочную литературу. Ввод и вывод информации на экране сопровождать соответствующими надписями.

|  |
| --- |
| 1. Дано четырехзначное число. Найти сумму первой и последней цифры этого числа.   int chislo, first, ten, hun, thous, result;  Console.WriteLine("Введите четырёхзначное число:");  chislo = Convert.ToInt32(Console.ReadLine());  first = chislo % 10;  ten = (chislo / 10) % 10;  hun = (chislo / 100) % 10;  thous = chislo / 1000;  result = first + thous;  Console.WriteLine($"Результат сложения первой и последней цифры - {result}");  Console.ReadKey(); |
| 1. Дано четырехзначное число. Переставить местами первую и последнюю цифры этого числа.   int chislo, first, ten, hun, thous, result;  Console.WriteLine("Введите четырёхзначное число:");  chislo = Convert.ToInt32(Console.ReadLine());  first = chislo % 10;  ten = (chislo / 10) % 10;  hun = (chislo / 100) % 10;  thous = chislo / 1000;    result = first \* 1000 + hun \* 100 + ten \* 10 + thous;  Console.WriteLine($"Ваше число - {chislo} \nЧисло, образованное перестановкой первой и последней цифры - {result}");  Console.ReadKey(); |
| 1. Дано натуральное число n < 99. Дописать к нему цифру k в конец и в начало.   Console.WriteLine("Введите целое число n, меньшее 99:");  int n = Convert.ToInt32(Console.ReadLine());  int first = n % 10;  int ten = (n / 10) % 10;  Console.WriteLine("Введите целое число k:");  int k = Convert.ToInt32(Console.ReadLine());  if (n < 99)  {  if (k < 10)  {  int result = k \* 1000 + n \* 10 + k;  Console.WriteLine($"n < 99 \nВаше полученное число - {result}");  }  else  {  Console.WriteLine($"Вы ввели k > 10. ({k}>10)");  }    }  else  {  Console.WriteLine($"Число {n} > 99. \nПолучить число не удалось.");  }  Console.ReadKey(); |
| 1. Дано натуральное число N из 6 цифр. Определить сумму цифр в первой половине числа (старшие разряды). Пример. N=123 456. Сумма составляет 1+2 + 3 = 6   int chislo, first, ten, hun, result, chislo\_preobr;  Console.WriteLine("Введите шестизначное число:");  chislo = Convert.ToInt32(Console.ReadLine());  chislo\_preobr = chislo / 1000;  first = chislo\_preobr % 10;  ten = (chislo\_preobr / 10) % 10;  hun = (chislo\_preobr / 100) % 10;  result = first + ten + hun;  Console.WriteLine($"Ваше число - {chislo}\nСумма цифр в первой половине числа - {result}");  Console.ReadKey(); |
| 1. Дано натуральное число N из 6 цифр. Определить сумму цифр во второй половине числа (младшие разряды). Пример. N=123 456. Сумма составляет 4+5 + 6= 15   int chislo, first, ten, hun, result;  Console.WriteLine("Введите шестизначное число:");  chislo = Convert.ToInt32(Console.ReadLine());  first = chislo % 10;  ten = (chislo / 10) % 10;  hun = (chislo / 100) % 10;  result = first + ten + hun;  Console.WriteLine($"Ваше число - {chislo}\nСумма цифр во второй половине числа - {result}");  Console.ReadKey(); |
| 1. Дано натуральное число из 5 цифр, получить новое число М, которое образуется путем исключения средней цифры исходного числа. Пример. N = 12345, М= 1245.   int chislo, first, second;  Console.WriteLine("Введите пятизначное число:");  chislo = Convert.ToInt32(Console.ReadLine());  first = chislo / 1000;  second = chislo % 100;  int result = first \* 100 + second;  Console.WriteLine($"Число, образованное путем исключения средней цифры числа {chislo} - {result}");  Console.ReadKey(); |
| 1. Дано трехзначное число. Определить сумму цифр данного числа.   int chislo, a, b, c, d;  Console.WriteLine("Введите трехзначное число:");  chislo = Convert.ToInt32(Console.ReadLine());  a = chislo % 10;  b = (chislo / 10) % 10;  c = (chislo / 100) % 10;  int result = a + b + c;  Console.WriteLine($"Ваше число - {chislo}\nЕдиницы - {a} \nДесятки - {b} \nСотни - {c} \nРезультат суммы цифр - {result}");  Console.ReadKey(); |
| 1. Дано трехзначное число. Получить новое число, полученное перестановкой цифр десятков и сотен   int chislo, a, b, c, d;  Console.WriteLine("Введите трехзначное число:");  chislo = Convert.ToInt32(Console.ReadLine());  a = chislo % 10;  b = (chislo / 10) % 10;  c = (chislo / 100) % 10;  int result = b \* 100 + c \* 10 + a;  Console.WriteLine($"Ваше число - {chislo}\nЕдиницы - {a} \nДесятки - {b} \nСотни - {c} \nРезультат перестановки десяток и сотен - {result}");  Console.ReadKey(); |
| 1. Дано трехзначное число. Получить новое число, полученное перестановкой цифр единиц и сотен.   int chislo, a, b, c, d;  Console.WriteLine("Введите трехзначное число:");  chislo = Convert.ToInt32(Console.ReadLine());  a = chislo % 10;  b = (chislo / 10) % 10;  c = (chislo / 100) % 10;  int result = a \* 100 + b \* 10 + c;  Console.WriteLine($"Ваше число - {chislo}\nЕдиницы - {a} \nДесятки - {b} \nСотни - {c} \nРезультат перестановки единиц и сотен - {result}");  Console.ReadKey(); |
| 1. Дано четырехзначное число. Определить сумму цифр данного числа.   int chislo, a, b, c, d;  Console.WriteLine("Введите четырехзначное число:");  chislo = Convert.ToInt32(Console.ReadLine());  a = chislo % 10;  b = (chislo / 10) % 10;  c = (chislo / 100) % 10;  d = chislo / 1000;  int summa = a + b + c + d;  Console.WriteLine($"Ваше число - {chislo}\nЕдиницы - {a} \nДесятки - {b} \nСотни - {c} \nТысячи - {d} ");  Console.WriteLine($"Сумма цифр - {summa} ");  Console.ReadKey(); |
| 1. Дано четырехзначное число. Получить новое число, полученное перестановкой цифр единиц и сотен.   int chislo, a, b, c, d;  Console.WriteLine("Введите четырехзначное число:");  chislo = Convert.ToInt32(Console.ReadLine());  a = chislo % 10;  b = (chislo / 10) % 10;  c = (chislo / 100) % 10;  d = chislo / 1000;  int newest = d \* 1000 + c \* 100 + a \* 10 + b;  Console.WriteLine($"Ваше число - {chislo}\nЕдиницы - {a} \nДесятки - {b} \nСотни - {c} \nТысячи - {d} ");  Console.WriteLine($"Число, полученное перестановкой цифр единиц и сотен - {newest} ");  Console.ReadKey(); |
| 1. Дано четырехзначное число. Получить новое число, полученное перестановкой цифр тысяч и сотен   int chislo, a, b, c, d;  Console.WriteLine("Введите четырехзначное число:");  chislo = Convert.ToInt32(Console.ReadLine());  a = chislo % 10;  b = (chislo / 10) % 10;  c = (chislo / 100) % 10;  d = chislo / 1000;  int newest = c \* 1000 + d \* 100 + b \* 10 + a;  Console.WriteLine($"Ваше число - {chislo}\nЕдиницы - {a} \nДесятки - {b} \nСотни - {c} \nТысячи - {d} ");  Console.WriteLine($"Число, полученное перестановкой цифр тысяч и сотен - {newest} ");  Console.ReadKey(); |
| 1. Дано четырехзначное число. Получить новое число, полученное перестановкой цифр единиц и сотен   int chislo, a, b, c, d;  Console.WriteLine("Введите четырехзначное число:");  chislo = Convert.ToInt32(Console.ReadLine());  a = chislo % 10;  b = (chislo / 10) % 10;  c = (chislo / 100) % 10;  d = chislo / 1000;  int newest = d \* 1000 + c \* 100 + a \* 10 + b;  Console.WriteLine($"Ваше число - {chislo}\nЕдиницы - {a} \nДесятки - {b} \nСотни - {c} \nТысячи - {d} ");  Console.WriteLine($"Число, полученное перестановкой цифр единиц и сотен - {newest} ");  Console.ReadKey(); |
| 1. Дано четырехзначное число. Получить новое число, полученное перестановкой цифр единиц и десятков   int number, units, dozens, hundreds, thousand, sum;  Console.WriteLine("Введите четырёхзначное число");  number = int.Parse(Console.ReadLine());  units = number % 10;  dozens = number / 10 % 10;  hundreds = number / 100 % 10;  thousand = number / 1000;  sum = thousand \* 1000 + hundreds \* 100 + dozens + units \* 10;  Console.WriteLine("Новое число, полученное перестановкой цифр единиц и десятков - {0}", sum); |
| 1. Дано четырехзначное число. Определить произведение цифр данного числа   int number, units, dozens, hundreds, thousand, proz;  Console.WriteLine("Введите четырёхзначное число");  number = int.Parse(Console.ReadLine());  units = number % 10;  dozens = number / 10 % 10;  hundreds = number / 100 % 10;  thousand = number / 1000;  proz = units \* dozens \* hundreds \* thousand; ;  Console.WriteLine("Новое число, полученное перестановкой цифр единиц и десятков - {0}", proz); |
| 1. Дано четырехзначное число. Найти сумму цифр десятков и единиц этого числа   int number, units, dozens, hundreds, thousand, sum;  Console.WriteLine("Введите четырёхзначное число");  number = int.Parse(Console.ReadLine());  units = number % 10;  dozens = number / 10 % 10;  hundreds = number / 100 % 10;  thousand = number / 1000;  sum = dozens + units;  Console.WriteLine("Сумма цифр единиц и десятков - {0}", sum); |
| 1. Дано четырехзначное число. Переставить местами вторую и последнюю цифры этого числа.   int number, units, dozens, hundreds, thousand, sum;  Console.WriteLine("Введите четырёхзначное число");  number = int.Parse(Console.ReadLine());  units = number % 10;  dozens = number / 10 % 10;  hundreds = number / 100 % 10;  thousand = number / 1000;  sum = thousand \* 1000 + hundreds + dozens \* 10 + units \* 100;  Console.WriteLine("Новое число, полученное перестановкой второй и последней цифры числа - {0}", sum); |
| 1. Дано трехзначное число. Дописать к нему цифру k в конец и в начало.   int n, k, s;  Console.WriteLine("Введите трёхзначное число");  n = int.Parse(Console.ReadLine());  Console.WriteLine("Введите цифру k");  k = int.Parse(Console.ReadLine());  s = k \* 10000 + n \* 10 + k;  Console.WriteLine($"Новое число {s}");  Console.ReadKey(); |

Контрольная работа №1

Целочисленная арифметика без цикла

Задание 1: Дано натуральное число N из 6 цифр. Определить произведение цифры сотен тысяч и цифры тысяч. Пример. N=123 456. Ответом будет 1\*3= 3.

int chislo, first, ten, hun, result, chislo\_preobr;

Console.WriteLine("Введите шестизначное число:");

chislo = Convert.ToInt32(Console.ReadLine());

chislo\_preobr = chislo / 1000; //первая половина вводимого числа

first = chislo\_preobr % 10; // сотни тысяч

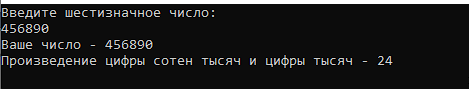
ten = (chislo\_preobr / 10) % 10; // десятки тысяч

hun = (chislo\_preobr / 100) % 10; // тысячи

result = first \* hun;

Console.WriteLine($"Ваше число - {chislo}\nПроизведение цифры сотен тысяч и цифры тысяч - {result}");

Console.ReadKey();



Задание 2: С начала суток прошло n секунд (n –целое). Определить сколько полных секунд прошло с начала очередной минуты.

Console.WriteLine("Введите число секунд, прошедших с начала суток:");

int n = Convert.ToInt32(Console.ReadLine());

int minute = n % 60;

Console.WriteLine($"С начала новой минуты прошло: {minute}");



Лабораторная работа №3

Тема: Операторы ветвления.

Цель: Научиться использовать операторы ветвления при выполнении расчетных проектов.

Задание 1. Разработать консольные приложения в соответствии с условием заданий. Решить по одной задаче из каждого раздела. Задачи выполняются в соответствии с номером по журналу.

Задание 1. 1

Вычислить значение функции. Значение x ввести с экрана. Результат вывести на экран.



double result;

Console.WriteLine("Введите значение x:");

int x = Convert.ToInt32(Console.ReadLine());

if (x > -3.5)

{

result = 4 \* Math.Pow(x, 2) + 2 \* x - 19;

Console.WriteLine($"Значение функции 1 - {result}");

}

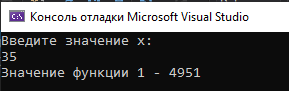
if (x < 3.5)

{

result = -(2 \* x / -(4 \* x) + 1);

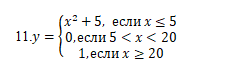
Console.WriteLine($"Значение функции 2 - {result}");

}



Задание 1.2

Вычислить значение функции. Значение x ввести с экрана. Результат вывести на экран. Использовать два условных оператора.



double result = 0;

Console.WriteLine("Введите значение x:");

int x = Convert.ToInt32(Console.ReadLine());

if (x <= 5)

{

result = Math.Pow(x, 2) + 5;

}

//if

else

{

if (x > 5 && x < 20)

{

result = 0;

}

//if

if (x >= 20)

{

result = 1;

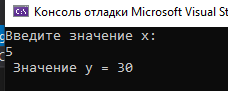
}

//if

}

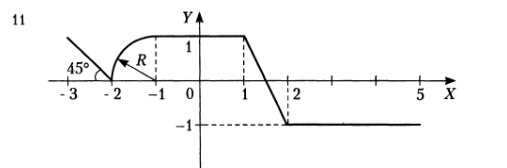
//else

Console.WriteLine($" Значение y = {result}");



Задание 2. Разработать консольные приложения в соответствии с условием заданий. Решить по одной задаче из каждого раздела. Задачи выполняются в соответствии с номером по журналу.

Задание 2.1. Написать программу, которая вычисляет значение функции по значению аргумента, введенного с клавиатуры. Функции заданы графически. Параметр R и значение аргумента вводятся с клавиатуры.



double y = 0;

Console.WriteLine("Введите значение x:");

double x = Convert.ToInt32(Console.ReadLine());

if (x >= -3 && x <= -2)

{

y = 4 \* x + 1;

}

if (x > -2 && x <= -1)

{

y = Math.Sqrt(1-Math.Pow((x-1), 2));

}

if (x > -1 && x <= 1)

{

y = 1;

}

if (x > 1 && x <= 2)

{

y =1.5 - 0.5 \* x;

}

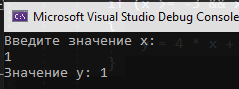
if (x > 2 && x <= 5)

{

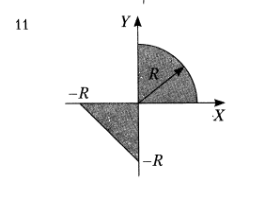
y = -1;

}

Console.WriteLine($"Значение y: {y}");



Задание 2.2.Написать программу, которая определяет, попадает ли точка с заданными координатами в область, закрашенную на рисунке серым цветом. Результат работы программы вывести в виде текстового сообщения. Значения координат точки вводить с клавиатуры.



Контрольная работа №2

Тема: Условный оператор

Задание 1. Дано натуральное число. Определить, оканчивается ли оно цифрой 7.

int a;

Console.WriteLine("Введите натуральное число:");

a = Convert.ToInt32(Console.ReadLine());

int first = a % 10;

if (first == 7)

{

Console.WriteLine("Число оканчивается на 7");

}

else

{

Console.WriteLine("Число не оканчивается на 7");

}

Console.ReadLine();

Задание 2.

double x, r;

double y = 0;

Console.WriteLine("Введите x:");

x = Convert.ToDouble(Console.ReadLine());

if (x >= -10 && x <=0)

{

y = -0.5 \* x - 3;

}

if (x > 0 && x <= 3)

{

y = Math.Sqrt(9 - Math.Pow(x, 2));

}

if (x > 3 && x <= 6)

{

y = Math.Sqrt(9 - Math.Pow((x - 6), 2));

}

Console.WriteLine($"y = {y}");

Лабораторная работа №4

Тема:Цикл с параметром

Задания для самостоятельной работы

1. Найти все двузначные числа, в которых есть цифра N или само число делится на N.

int n;

Console.WriteLine("Введите число n");

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

for (int i = 10; i < 99; i++)

{

if ((i / 10 % 10 == n) || (i % 10 == n) || (i / n == 0))

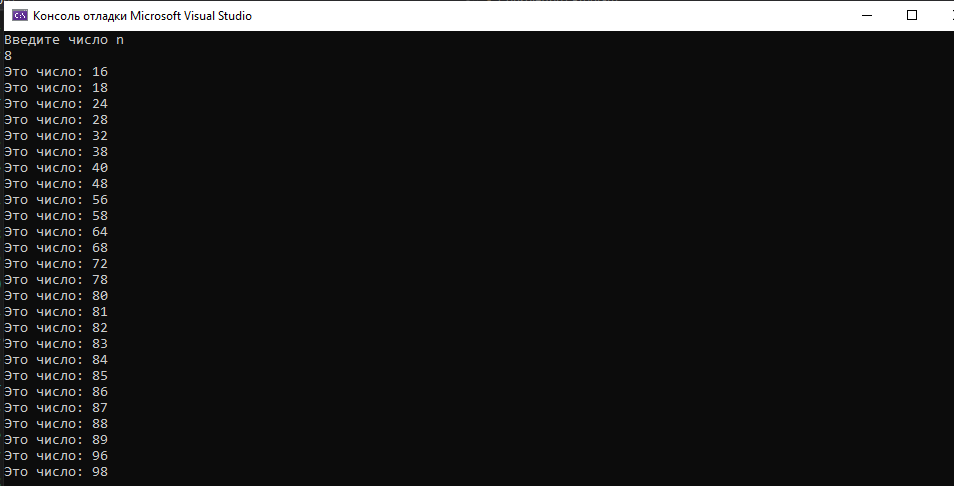
{

Console.WriteLine($"это число: {i}");

}

}

Console.ReadKey();



2. Определить количество трехзначных натуральных чисел, сумма цифр которых равна заданному числу N.

double n;

int count = 0;

Console.WriteLine("Введите число n");

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

for (int i = 100; i < 999; i++)

{

int one = i % 10;

int dec = (i / 10) % 10;

int hun = i / 100;

int sum = one + dec + hun;

if (sum == n)

{

Console.WriteLine($"Нашли число: {i}");

count = count + 1;

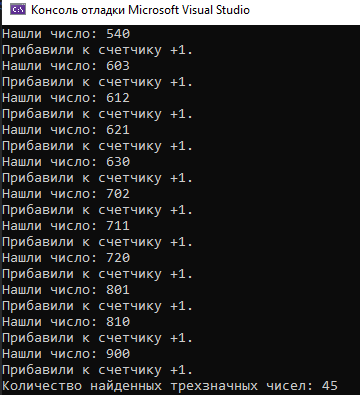
Console.WriteLine($"Прибавили к счетчику +1.");

}

}//for

Console.WriteLine($"Количество найденных трехзначных чисел: {count}");

Console.ReadKey();



3. Составить программу вычисления суммы кубов чисел от 25 до 55.

double sum = 0;

for (int i = 25; i < 56; i++)

{

sum += i \* i \* i;

}

Console.WriteLine($"Сумма кубов чисел от 25 до 55: {sum}");

Console.ReadKey();



4. Среди двузначных чисел найти те, сумма квадратов цифр которых делится на 13.

double one, dec, sum;

for (int i = 10; i < 100; i++)

{

one = i % 10;

dec = i / 10;

sum = Math.Pow(one, 2) + Math.Pow(dec, 2);

if (sum % 13 == 0)

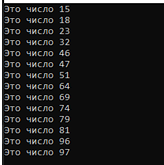
{

Console.WriteLine($"Это число {i}");

}

}

Console.ReadKey();



5. Написать программу поиска двузначных чисел, таких, что если к сумме цифр этого числа прибавить квадрат этой суммы, то получится это число.

double one, dec, sum, sumK;

for (int i = 10; i < 100; i++)

{

one = i % 10;

dec = i / 10;

sum = one + dec;

sumK = sum + Math.Pow(sum, 2);

if (sumK == i)

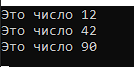
{

Console.WriteLine($"Это число {i}");

}

}

Console.ReadKey();



6. Квадрат трехзначного числа оканчивается тремя цифрами, которые как раз и составляют это число. Написать программу поиска таких чисел.

double square, n;

for (int i = 100; i < 1000; i++)

{

square = Math.Pow(i, 2);

n = square % 1000;

if (n == i)

{

Console.WriteLine($"Это число {i}");

}

}

Console.ReadKey();



7. Написать программу поиска четырехзначного числа, которое при делении на 133 дает в остатке 125, а при делении на 134 дает в остатке 111.

int count = 0;

for (int i = 1000; i < 10000; i++)

{

if ((i % 133 == 125) && (i % 134 == 111))

{

Console.WriteLine($"Это число {i}");

count += 1;

}

}

Console.WriteLine($"Количество подсчитанных чисел: {count}");

Console.ReadKey();



8. Найти сумму положительных нечетных чисел, меньших 100.

double sum = 0;

for (int i = 1; i < 100; i++)

{

if (i % 2 == 1)

{

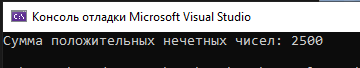
sum += i;

}

}

Console.WriteLine($"Сумма положительных нечетных чисел: {sum}");

Console.ReadKey();



1. Найти сумму целых положительных чисел из промежутка от А до В, кратных 4 (значения переменных А и В вводятся с клавиатуры).

int a, b, sum = 0;

Console.Write("Введите значение промежутка А (начало):");

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

Console.Write("Введите значение промежутка Б (конец):");

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

for (int i = a; i < b; i++)

{

if (i % 4 == 0)

{

sum += i;

}

}

Console.WriteLine($"Сумма целых положительных чисел промежутка от А до В, кратных 4 = {sum}");

Console.ReadKey();



10. Найти сумму целых положительных чисел, больших 20, меньших 100, кратных 3 и заканчивающихся на 2, 4 или 8

int sum = 0;

for (int i = 21; i < 100; i++)

{

if (i % 3 == 0 && (i % 10 == 2 || i % 10 == 4 || i % 10 == 8))

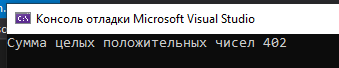
{

sum += i;

}

}

Console.WriteLine($"Сумма целых положительных чисел {sum}");



11. В трехзначном числе зачеркнули первую цифру слева, когда полученное двузначное число умножили на 7, то получили данное число. Найти это число.

int n;

for (int i = 100; i < 1000; i++)

{

n = i % 100;

if (i == 7 \* n)

{

Console.WriteLine($"Это число {i}");

}

}



12. Сумма цифр трехзначного числа кратна 7, само число также делится на 7. Найти все такие числа.

int sum, one, dec, hun;

for (int i = 100; i < 1000; i++)

{

one = i % 10;

dec = (i / 10) % 10;

hun = i / 100;

sum = one + dec + hun;

if (sum % 7 == 0 && i % 7 == 0)

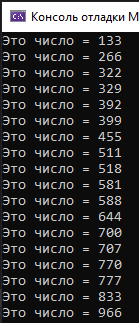
{

Console.WriteLine($"Это число = {i}");

}

}

Console.ReadKey();



13. Среди четырехзначных чисел выбрать те, у которых все четыре цифры различны.

int one, dec, hun, th;

for (int i = 1000; i < 10000; i++)

{

one = i % 10;

dec = (i / 10) % 10;

hun = (i / 100) % 10;

th = i / 1000;

if (th != one && th != dec && th != hun && hun != one && hun != dec && dec != one)

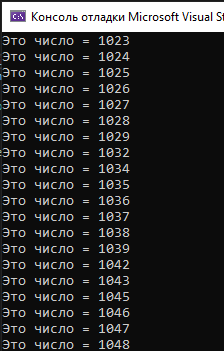
{

Console.WriteLine($"Это число = {i}");

}

}

Console.ReadKey();



14. Среди двузначных чисел найти те, сумма цифр которых равна n(0<n<18) и число делится без остатка на число q.

int sum = 0, one, dec, q;

Console.WriteLine("Введите число q");

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

for (int i = 10; i < 100; i++)

{

one = i % 10;

dec = i / 10;

sum = one + dec;

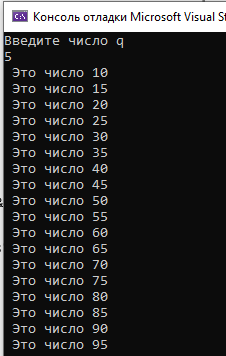
if ((0 < sum) && (18 > sum) && (i % q == 0))

{

Console.WriteLine($" Это число {i}");

}

}



15. Одна штука некоторого товара стоит 20,4 руб. Напечатать таблицу стоимости 2, 3, ..., 20 штук этого товара.

double k = 20.4;

double proz;

Console.WriteLine("Одна штука некоторого товара стоит 20,4 руб");

for (int i = 2; i < 21; i++)

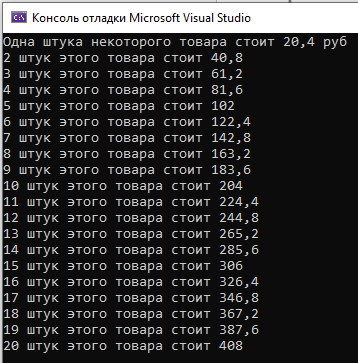
{

proz = k \* i;

Console.WriteLine($"{i} штук этого товара стоит {proz:#.##}");

}

Console.ReadKey();



16. Напечатать таблицу умножения на 7:  
1 х 7 = 7  
2 х 7 = 14  
...  
9 х 7 = 63

int k = 7, proz;

Console.WriteLine("Таблица умножения на 7");

for (int i = 1; i < 10; i++)

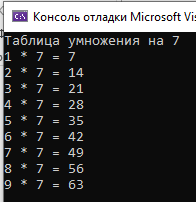
{

proz = k \* i;

Console.WriteLine($"{i} \* {k} = {proz}");

}

Console.ReadKey();



17. Рассчитать значения y для значений х, равных 4, 5, ..., 28:

double y, t;

for (int x = 4; x < 29; x++)

{

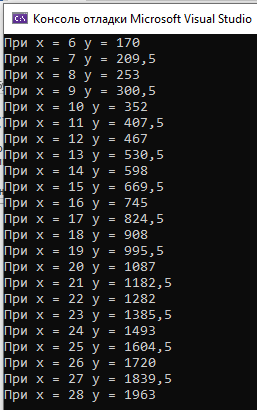
t = x + 2;

y = 2 \* Math.Pow(t, 2) + 5.5 \* t - 2;

Console.WriteLine($"При x = {x} y = {y}");

}

Console.ReadKey();



18. Дано четырехзначное число n. Выбросить из записи числа n цифры 0 и 5, оставив прежним порядок остальных цифр. Например, из числа 1509 должно получиться 19.

int n, k = 0, l = 1, m;

Console.WriteLine("Введите четырёхзначное число ");

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

m = n;

for (int i = 0; i < 4; i++)

{

if (n % 10 != 0 && n % 10 != 5)

{

k = k + n % 10 \* l;

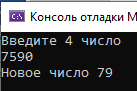
l \*= 10;

}

n /= 10;

}

Console.WriteLine($"Новое число {k}");



1. Натуральное число из n цифр является числом Армстронга, если сумма его цифр, возведенных в n-ю степень, равна самому числу (например, 153=13+53+33). Получить все числа Армстронга, состоящие из трех и четырех цифр.

int one, dec, hun, th, m;

for (int i = 100; i < 9999; i++)

{

m = i;

one = i % 10;

dec = (i / 10) % 10;

hun = (i / 100) % 10;

th = i / 1000;

if ((Math.Pow(one, 3) + Math.Pow(dec, 3) + Math.Pow(hun, 3) + Math.Pow(th, 3) == i))

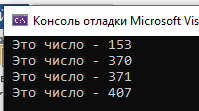
{

Console.WriteLine($" Это число - {i}");

}

}

}



1. Дана последовательность из 20 целых чисел. Определить количество чисел в наиболее длинной под последовательности из подряд идущих нулей.

21. Дано натуральное число. Найти все его делители и их сумму.

int n, k = 0, sum = 0;

Console.WriteLine("Введите число ");

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

for (int i = 2; i <= n / 2; i++)

{

if (n % i == 0)

{

Console.WriteLine($" Это делитель {i}");

k++;

sum += i;

}

}

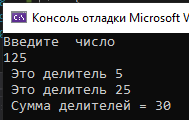
if (k == 0)

{

Console.WriteLine($" У этого числа есть только два делителя 1 и {n}");

}

Console.WriteLine($" Сумма делителей = {sum}");



Контрольная работа №3

Цикл с параметром

Необходимо решить три задачи.

Из первого задания необходимо решить номер по варианту и номер по варианту+10. Если такого варианта нет, отсчет продолжается с первого варианта. При оценивании работы будет учитываться вид ввода исходных данных и вывод результата. Должны присутствовать поясняющие надписи при вводе и выводе.

Из второго задания необходимо решить номер по варианту.

Работа рассчитана на 60 мин.

1. Среди трехзначных чисел найти те, сумма квадратов цифр которых делится на 18.

double one, dec, sum, hun;

for (int i = 100; i < 1000; i++)

{

one = i % 10;

dec = (i / 10) % 10;

hun = i / 100;

sum = Math.Pow(one, 2) + Math.Pow(dec, 2) + Math.Pow(hun, 2);

if (sum % 18 == 0)

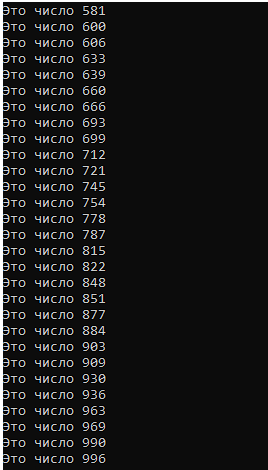
{

Console.WriteLine($"Это число {i}");

}

}

Console.ReadKey();



1. Найти количество целых положительных чисел из промежутка от А до В, кратных 3 и оканчивающихся на 1 (значения переменных А и В вводятся с клавиатуры).

int i;

int one = 0;

Console.WriteLine("Введите начало отрезка А (целое и положительное число):");

int otr\_A = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите начало отрезка B (целое и положительное число):");

int otr\_B = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите шаг:");

int step = Convert.ToInt32(Console.ReadLine());

for (i = otr\_A; i <= otr\_B; i += step)

{

one = i % 10;

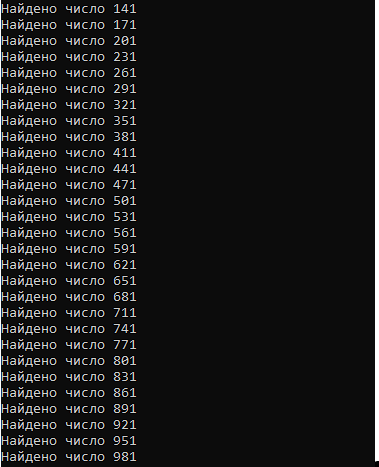
if (one == 1 && i % 3 == 0) {

Console.WriteLine($"Найдено число {i}");

}//if

}//for

Console.ReadKey();



1. Составить программу для вычисления значений функции F(x) на отрезке [a,b] с шагом h. Результат представить в виде таблицы, первый столбец – значение аргумента, второй - соответствующее значение функции. Границы отрезка и шаг ввести с экрана. При написании формулы для вычисления функции использовать методы класса Math.

F(x)=.

int i;

double fx = 0;

Console.WriteLine("Введите начало отрезка А:");

int otr\_A = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите начало отрезка B:");

int otr\_B = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите шаг:");

int step = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Функция: F(x) = x - sin(x)");

for (i = otr\_A; i <= otr\_B; i += step)

{

fx = i - Math.Sin(i);

if (i < 10)

{

Console.WriteLine($"| x = {i} | F(x) = {fx:f2} |");

}

if(i > 10 && i < 99)

{

Console.WriteLine($"| x = {i}| F(x) = {fx:f2}|");

}

if (i == 99)

{

Console.WriteLine($"| x = {i}| F(x) ={fx:f2}|");

}

if (i > 99)

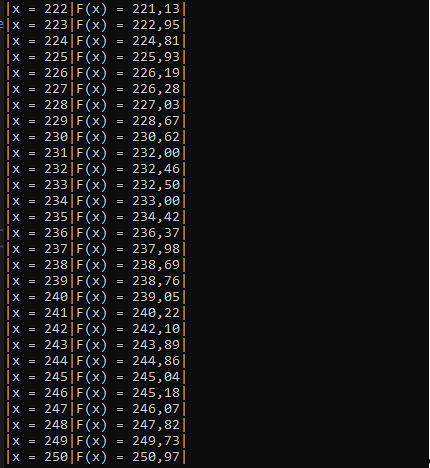
{

Console.WriteLine($"|x = {i}|F(x) = {fx:f2}|");

}

}

Console.ReadKey();

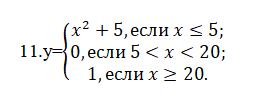


Лабораторная работа 5-6

Тема: Циклы с постусловием и предусловием.

Задание 1. Задачу решить двумя способами с использованием двух видов цикла.

Постройте таблицу значений функции y=f(x) для



double y = 0;

Console.WriteLine("Введите начало отрезка А:");

int otr\_A = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите начало отрезка B:");

int otr\_B = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите шаг:");

int step = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Рассчет y циклом while в табличном виде:");

while (otr\_A <= otr\_B)

{

if (otr\_A <= 5)

{

y = Math.Pow(otr\_A, 2) + 5;

if (otr\_A >= 3 && otr\_A <= 5)

{

Console.WriteLine($"| x = {otr\_A} | y = {y}|");

}

else

{

Console.WriteLine($"| x = {otr\_A} | y = {y} |");

}

}

if (otr\_A < 20 && otr\_A > 5)

{

y = 0;

if (otr\_A >= 10)

{

Console.WriteLine($"| x = {otr\_A}| y = {y} |");

}

else

{

Console.WriteLine($"| x = {otr\_A} | y = {y} |");

}

}

if (otr\_A >= 20)

{

y = 1;

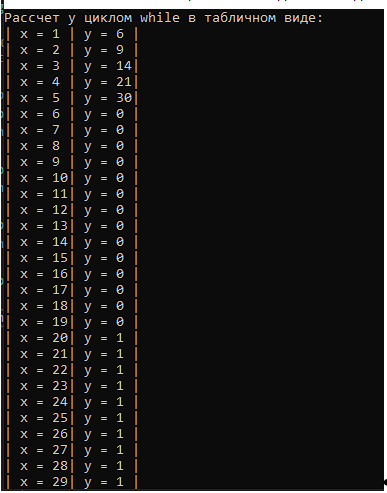
Console.WriteLine($"| x = {otr\_A}| y = {y} |");

}

otr\_A += step;

}//while

Console.ReadKey();



double y = 0;

Console.WriteLine("Введите начало отрезка А:");

int otr\_A = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите начало отрезка B:");

int otr\_B = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите шаг:");

int step = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Рассчет y циклом do while в табличном виде:");

do

{

if (otr\_A <= 5)

{

y = Math.Pow(otr\_A, 2) + 5;

if (otr\_A >= 3 && otr\_A <= 5)

{

Console.WriteLine($"| x = {otr\_A} | y = {y}|");

}

else

{

Console.WriteLine($"| x = {otr\_A} | y = {y} |");

}

}

if (otr\_A < 20 && otr\_A > 5)

{

y = 0;

if (otr\_A >= 10)

{

Console.WriteLine($"| x = {otr\_A}| y = {y} |");

}

else

{

Console.WriteLine($"| x = {otr\_A} | y = {y} |");

}

}

if (otr\_A >= 20)

{

y = 1;

Console.WriteLine($"| x = {otr\_A}| y = {y} |");

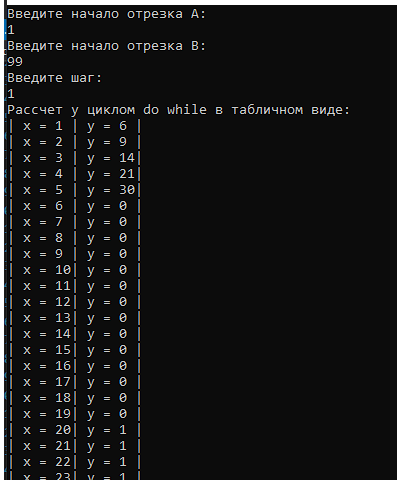
}

otr\_A += step;

}

while (otr\_A <= otr\_B);

Console.ReadKey();



Задание 2. Задачу решить с использованием двух видов цикла.

Составить программу для вычисления значений функции F(x) на отрезке



double fx = 0;

Console.WriteLine("Введите начало отрезка А:");

int otr\_A = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите начало отрезка B:");

int otr\_B = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите шаг:");

int step = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Решение задачи с циклом while");

while (otr\_A <= otr\_B)

{

fx = Math.Sqrt(otr\_A) \* Math.Pow(Math.Cos(otr\_A), 2);

if (otr\_A < 10)

{

Console.WriteLine($"| x = {otr\_A} | F(x) = {fx:f2}|");

}

if (otr\_A > 10 && otr\_A < 99)

{

Console.WriteLine($"| x = {otr\_A}| F(x) = {fx:f2}|");

}

if (otr\_A == 99)

{

Console.WriteLine($"| x = {otr\_A}| F(x) = {fx:f2}|");

}

if (otr\_A > 99)

{

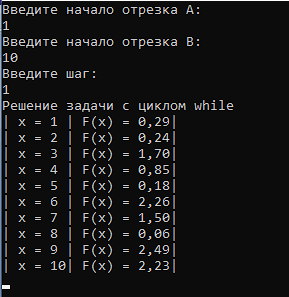
Console.WriteLine($"|x = {otr\_A}|F(x) = {fx:f2}|");

}

otr\_A += step;

}

Console.ReadKey();



double fx = 0;

Console.WriteLine("Введите начало отрезка А:");

int otr\_A = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите начало отрезка B:");

int otr\_B = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите шаг:");

int step = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Решение задачи с циклом do while");

do

{

fx = Math.Sqrt(otr\_A) \* Math.Pow(Math.Cos(otr\_A), 2);

if (otr\_A < 10)

{

Console.WriteLine($"| x = {otr\_A} | F(x) = {fx:f2}|");

}

if (otr\_A >= 10 && otr\_A < 99)

{

Console.WriteLine($"| x = {otr\_A}| F(x) = {fx:f2}|");

}

if (otr\_A == 99)

{

Console.WriteLine($"| x = {otr\_A}| F(x) = {fx:f2}|");

}

if (otr\_A > 99)

{

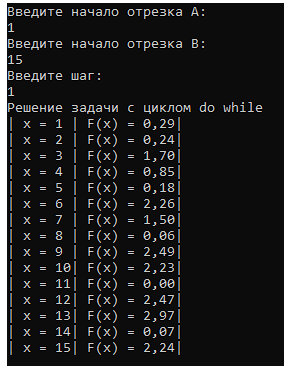
Console.WriteLine($"|x = {otr\_A}|F(x) = {fx:f2}|");

}

otr\_A += step;

} while (otr\_A <= otr\_B);

Console.ReadKey();



Задание 3. Решить все задачи двумя способами с использованием двух видов цикла while и do while

1. Напечатать "столбиком":

а) все целые числа от 20 до 35;

int i = 20;

int k = 36;

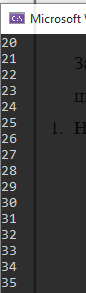
while (i < k)

{

Console.WriteLine($"{i}");

i++;

}



б) квадраты всех целых чисел от 10 до b(значение bвводится с клавиатуры; b *>* 10);

int i = 10;

Console.WriteLine("Введите значение b>10:");

int b = Convert.ToInt32(Console.ReadLine());

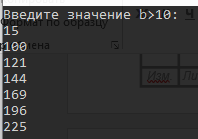
while (i <= b)

{

Console.WriteLine($"{i\*i}");

i++;

}



г) все целые числа от aдо b(значения aи bвводятся с клавиатуры; b > a).

Console.WriteLine("Введите значение a:");

int a = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите значение b:");

int b = Convert.ToInt32(Console.ReadLine());

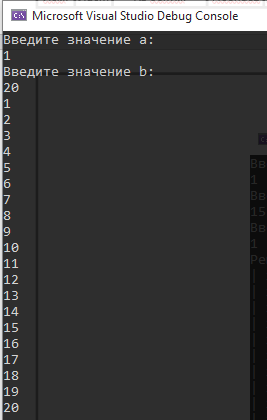
while (a <= b)

{

Console.WriteLine($"{a}");

a++;

}



1. Одна штука некоторого товара стоит 20,4 руб. Напечатать таблицу стоимости 2, 3, 20 штук этого товара.

double k = 20.4;

double proz;

Console.WriteLine("Одна штука некоторого товара стоит 20,4 руб");

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

int i = 1;

while ( i < 21)

{

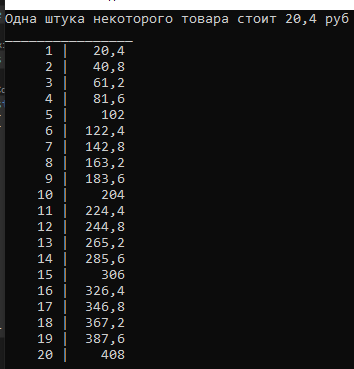
proz = k \* i;

Console.WriteLine($"{i,6} | {proz,6:#.##}");

i++;

}

Console.ReadKey();



1. Напечатать таблицу умножения на число *п* (значение *п* вводится с клавиатуры; 1 < n *<* 9).

1 х n =…

2xn=…

…

9 х n = …

Console.Write("Введите число ");

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

Console.WriteLine($"Таблица умножения на {n}");

int k = n, proz, i = 1;

while ( i < 10 )

{

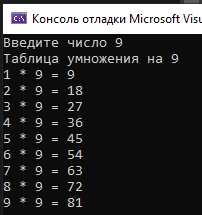
proz = k \* i;

Console.WriteLine($"{i} \* {k} = {proz}");

i++;

}

Console.ReadKey();



1. Найти все двузначные числа, в которых есть цифра N или само число делится на N.

int n, i = 10;

Console.WriteLine("Введите число n");

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

while (i < 100)

{

if ((i / 10 == n) || (i % 10 == n))

{

Console.WriteLine($"Число {i} содержит число {n}");

}

if (i % n == 0)

{

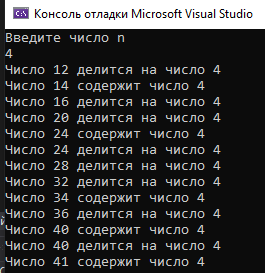
Console.WriteLine($"Число {i} делится на число {n}");

}

i++;

}

Console.ReadKey();



1. Определить количество трехзначных натуральных чисел, сумма цифр которых равна заданному числу N.

int n, k = 0, i = 100;

Console.WriteLine("Введите число ");

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

while (i < 1000)

{

int one = i % 10;

int dec = i / 10 % 10;

int hun = i / 100;

if (one + dec + hun == n)

{

k++;

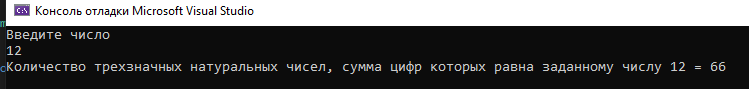
}

i++;

}

Console.WriteLine($"Количество трехзначных натуральных чисел, сумма цифр которых равна заданному числу {n} = {k}");

Console.ReadKey();



1. Квадрат трехзначного числа оканчивается тремя цифрами, которые как раз и составляют это число. Написать программу поиска таких чисел.

double kon;

int i = 100;

while (i < 1000)

{

kon = Math.Pow(i, 2) % 1000;

if (i == kon)

{

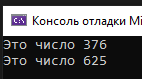
Console.WriteLine($"Это число {i} ");

}

i++;

}

Console.ReadKey();



1. Написать программу поиска четырехзначного числа, которое при делении на 133 дает в остатке 125, а при делении на 134 дает в остатке 111.

int i = 1000;

while (i < 10000)

{

if ((i % 133 == 125) && (i % 134 == 111))

{

Console.WriteLine($"Это число {i} ");

}

i++;

}

Console.ReadKey();



1. Найти сумму целых положительных чисел из промежутка от А до В, кратных 4 (значения переменных А и В вводятся с клавиатуры).

int b, a, sum = 0;

Console.WriteLine("Введите начальное ограничение a");

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

Console.WriteLine("Введите конечное ограничение b");

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

while (a <= b)

{

if (a % 4 == 0)

{

sum = sum + a;

}

a++;

}

Console.WriteLine($"Сумма чисел от a до b = {sum}");

Console.ReadKey();

1. В трехзначном числе зачеркнули первую цифру слева, когда полученное двузначное число умножили на 7, то получили данное число. Найти это число.

int hun, i = 100;

while (i < 1000)

{

hun = i % 100;

if (i == 7 \* hun)

{

Console.WriteLine($"Это число = {i}");

}

i++;

}

Console.ReadKey();



1. Среди четырехзначных чисел выбрать те, у которых все четыре цифры различны.

int one, dec, hun, th, i = 1000;

while (i < 10000)

{

one = i % 10;

dec = (i / 10) % 10;

hun = (i / 100) % 10;

th = i / 1000;

if (th != one && th != dec && th != hun && hun != one && hun != dec && dec != one)

{

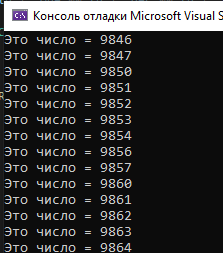
Console.WriteLine($"Это число = {i}");

}

i++;

}

Console.ReadKey();



1. Дано четырехзначное число n. Выбросить из записи числа n цифры 0 и 5, оставив прежним порядок остальных цифр. Например, из числа 1509 должно получиться 19.

int n, k = 0, l = 1, m;

Console.WriteLine("Введите четырёхзначное число ");

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

m = n;

int i = 0;

while (i < 4)

{

if (n % 10 != 0 && n % 10 != 5)

{

k = k + n % 10 \* l;

l \*= 10;

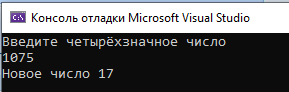
}

n /= 10;

i++;

}

Console.WriteLine($"Новое число {k}");



1. Натуральное число из n цифр является числом Армстронга, если сумма его цифр, возведенных в n-ю степень, равна самому числу (например, 153=13+53+33). Получить все числа Армстронга, состоящие из трех и четырех цифр.

int one, dec, hun, th, m, i = 1001;

while (i > 1000 && i < 10000)

{

m = i;

one = i % 10;

dec = (i / 10) % 10;

hun = (i / 100) % 10;

th = i / 1000;

if ((Math.Pow(one, 4) + Math.Pow(dec, 4) + Math.Pow(hun, 4) + Math.Pow(th, 4) == i))

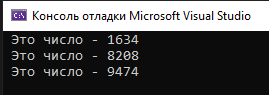
{

Console.WriteLine($" Это число - {i}");

}

i++;

}



1. Дано натуральное число. Найти все его делители и их сумму.

int one, dec, hun, m, i = 101;

while (i > 100 && i < 1000)

{

m = i;

one = i % 10;

dec = (i / 10) % 10;

hun = i / 100;

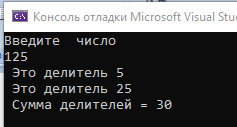
if ((Math.Pow(one, 3) + Math.Pow(dec, 3) + Math.Pow(hun, 3) == i))

{

Console.WriteLine($" Это число - {i}");

}

i++;



Задание 4. Решить все задачи. При решении задач используйте один вид цикла while или do while. Какой именно решите сами.

1. Дано целое число N (> 0). Используя операции деления нацело и взятия остатка от деления, найти количество и сумму его цифр.

int n, k = 0, sum = 0, s;

Console.WriteLine("Введите число n>0");

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

while (n > 0)

{

s = n % 10;

n /= 10;

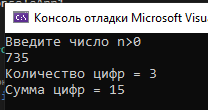
sum = s + sum;

k++;

}

Console.WriteLine($"Количество цифр = {k}");

Console.WriteLine($"Сумма цифр = {sum}");



1. Дано целое число N (> 0). С помощью операций деления нацело и взятия остатка от деления определить, имеется ли в записи числа N цифра «2». Если

имеется, то вывести true, если нет — вывести false.

Console.WriteLine("Введите число n>0");

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

int m = n, M = n, k = 0;

while (m > 0)

{

m %= 10;

if (m == 2)

{

k++;

}

m = M;

M /= 10;

}

if (k > 0)

{

Console.WriteLine("True");

}

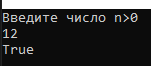
else

{

Console.WriteLine("False");

}

Console.ReadKey();



1. Найти минимальное число, большее 300, которое нацело делится на 19.

int m = 300;

while (m >= 300 && m%19 != 0)

{

m++;

if ( m % 19 == 0)

{

Console.WriteLine($"Минимальное число которое > 300 и делится ны 19 = {m}");

}

}

Console.ReadKey();



1. Приписать по 1 в начало и в конец записи числа n. Например, было n=3456, стало n=134561.

int n, k, x = 1, s;

Console.Write("Введите число n - ");

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

k = n;

do

{

n /= 10;

x \*= 10;

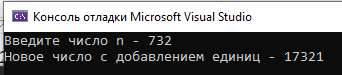
}

while (n > 0);

s = (x + k)\*10 + 1;

Console.WriteLine($"Новое число с добавлением единиц - {s}");

Console.ReadKey();



1. Поменять местами первую и последнюю цифры числа. Например, из числа 8547 должно быть получено число 7548.

int chislo, first, ten, hun, thous, result;

while (true) {

Console.WriteLine("Введите четырёхзначное число:");

chislo = Convert.ToInt32(Console.ReadLine());

first = chislo % 10;

ten = (chislo / 10) % 10;

hun = (chislo / 100) % 10;

thous = chislo / 1000;

result = first \* 1000 + hun \* 100 + ten \* 10 + thous;

Console.WriteLine($"Ваше число - {chislo} \nЧисло, образованное перестановкой первой и последней цифры - {result}");

Console.WriteLine("Завершить работу? (Да, Нет)");

string end = Console.ReadLine();

if (end.Equals("Да") == true) {

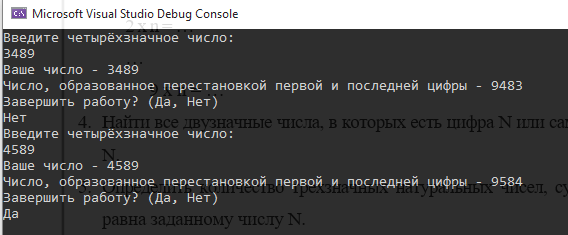
break;

}

else{

continue;

}



1. Приписать к исходному числу n такое же число. Например, из числа 1903 должно быть получено число 19031903.

string result;

while (true) {

Console.WriteLine("Введите четырёхзначное число:");

string chislo = Convert.ToString(Console.ReadLine());

result = chislo + chislo;

Console.WriteLine($"{result}");

Console.WriteLine("Завершить работу? (Да, Нет)");

string end = Console.ReadLine();

if (end.Equals("Да") == true) {

break;

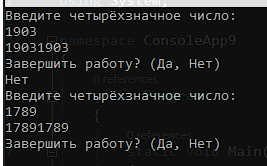
}

else{

continue;

}

}



1. Определить, является ли заданное число степенью 3.

while (true) {

Console.WriteLine("Введите число:");

int chislo = Convert.ToInt32(Console.ReadLine());

int i = 1;

while (i < chislo) {

i = i \* 3;

if (i == chislo)

{

Console.WriteLine($"Число {chislo} является степенью 3");

break;

}

}

Console.WriteLine("Завершить работу? (Да, Нет)");

string end = Console.ReadLine();

if (end.Equals("Да") == true) {

break;

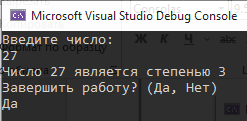
}

else{

continue;

}

}



1. Составить программу, проверяющую, является ли заданное натуральное число палиндромом, то есть таким, десятичная запись которого читается одинаково слева направо и справа налево.

while (true) {

Console.WriteLine("Введите число:");

int chislo = Convert.ToInt32(Console.ReadLine());

int i = 0;

int tmp = chislo;

while (tmp > 0) {

i = i \* 10 + tmp % 10;

tmp /= 10;

if (i == chislo)

{

Console.WriteLine($"Число {chislo} является палиндромом");

break;

}

}

Console.WriteLine("Завершить работу? (Да, Нет)");

string end = Console.ReadLine();

if (end.Equals("Да") == true) {

break;

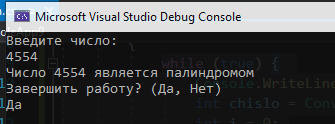
}

else{

continue;

}

}



1. Определить является ли введенное целое число простым.

Console.Write("Введите число n - ");

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

int b = 1, k = 0;

while (b <= n)

{

if (n % b == 0)

{

k++;

}

b++;

}

if (k == 2)

{

Console.WriteLine($"{n} - это простое число");

}

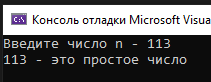
else

{

Console.WriteLine($"{n} это не простое число");

}

Console.ReadKey();



1. Дано целое число N (> 1). Найти наименьшее целое число K, при котором выполняется неравенство 3K > N.

while (true) {

Console.WriteLine("Введите число:");

int n = Convert.ToInt32(Console.ReadLine());

int k = 1;

while (3\*k <= n) {

k++;

}

Console.WriteLine($"Найти наименьшее целое число K, при котором выполняется неравенство 3K > N: {k}");

Console.WriteLine("Завершить работу? (Да, Нет)");

string end = Console.ReadLine();

if (end.Equals("Да") == true) {

break;

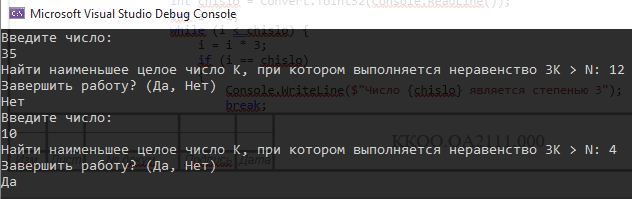
}

else{

continue;

}

}



1. Дано целое число N (> 1). Вывести наименьшее из целых чисел K, для которых сумма 1 + 2 + … + K будет больше или равна N, и саму эту сумму.

Console.Write("Введите N:");

int n = Convert.ToInt32(Console.ReadLine());

int k = 0;

int sum = 0;

while (sum <= n)

{

k++;

sum = ((1 + k)/2) \* k;

if (sum >= n)

{

sum = sum - k;

k--;

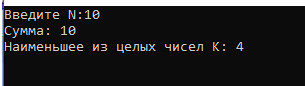
Console.WriteLine($"Сумма: {sum} \nНаименьшее из целых чисел K: {k}");

break;

}

}

Console.ReadKey();



1. Дано число A (> 1). Вывести наименьшее из целых чисел K, для которых сумма 1 + 1/2 + … + 1/K будет больше A, и саму эту сумму.

while (true) {

Console.WriteLine("Введите число:");

double a = Convert.ToDouble(Console.ReadLine());

int k = 0;

double tmp = 0;

while (tmp <= a) {

k++;

tmp = tmp + 1 / (double)k;

}

Console.WriteLine($"Наименьшее целое число K: {k}");

Console.WriteLine("Завершить работу? (Да, Нет)");

string end = Console.ReadLine();

if (end.Equals("Да") == true) {

break;

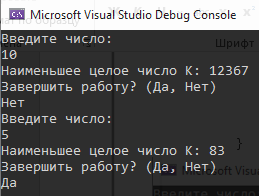
}

else{

continue;

}

}



1. Дано целое число N (> 0). Используя операции деления нацело и взятия остатка от деления, вывести все его цифры, начиная с самой правой (разряда единиц).

while (true) {

Console.WriteLine("Введите число:");

int n = Convert.ToInt32(Console.ReadLine());

while (n > 0) {

Console.WriteLine($"{n%10}");

n /= 10;

}

Console.WriteLine("Завершить работу? (Да, Нет)");

string end = Console.ReadLine();

if (end.Equals("Да") == true) {

break;

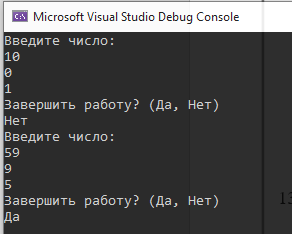
}

else{

continue;

}

}



1. Совершенным числом называется число, равное сумме всех своих делителей, меньших, чем оно само. Например: 6=1+2+3, 28=1+2+4+7+14. Древним грекам были известны только четыре первых числа. Составить программу, проверяющую, является ли заданное натуральное число совершенным.

while (true) {

Console.WriteLine("Введите число:");

int n = Convert.ToInt32(Console.ReadLine());

int i = 1;

int sum = 0;

while (i < n) {

if (n % i == 0)

{

sum += i;

}

if (n == sum){

Console.WriteLine($"Число {n} совершенное");

break;

}

i++;

}

Console.WriteLine("Завершить работу? (Да, Нет)");

string end = Console.ReadLine();

if (end.Equals("Да") == true) {

break;

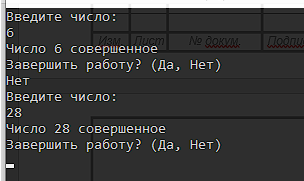
}

else{

continue;

}

}



1. Автоморфным называется такое число, которое равно послед­ним цифрам своего квадрата. Например: 52=25, 252=625. Оче­видно, что автоморфные числа должны оканчиваться либо на 1, либо на 5, либо на 6. Составить программу нахождения автоморфных чисел (с учетом приведенного факта), не превышающих значения 999.

int a = 1;

while (a <= 999)

{

if (a == Math.Pow(a, 2) % 10 || a == Math.Pow(a, 2) % 100 || a == Math.Pow(a, 2) % 1000)

{

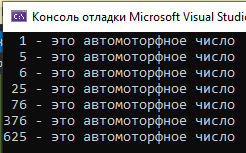
Console.WriteLine($"{a,3} - это автомоторфное число");

}

a++;

}

Console.ReadKey();



1. Кубические автоморфные числа равны последним цифрам своих кубов. Например: 63=216. Верно ли, что и такие числа должны оканчиваться либо на 1, либо на 5, либо на 6? С учетом этого факта составить программу нахождения двузначных, трехзначных кубических автоморфных чисел.

int a = 1;

while (a <= 999)

{

if (a == Math.Pow(a,3) % 10 || a == Math.Pow(a,3) % 100 || a == Math.Pow(a,3) % 1000)

{

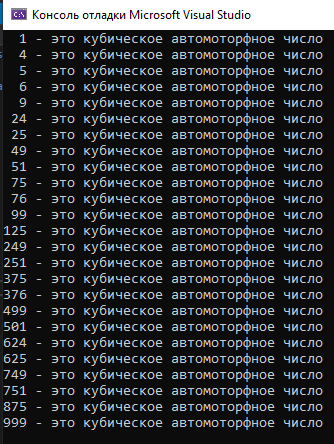
Console.WriteLine($"{a,3} - это кубическое автомоторфное число");

}

a++;

}

Console.ReadKey();



Контрольная работа №4

Циклы с предусловием и постусловием

Необходимо решить три задачи.

Из задания 1 необходимо решить номер по варианту.

Из задания 2 необходимо решить номер по варианту и номер по варианту+10. Если такого варианта нет, отсчет продолжается с первого варианта. При оценивании работы будет учитываться вид ввода исходных данных и вывод результата. Должны присутствовать поясняющие надписи при вводе и выводе.

Работа рассчитана на 60 мин.

Критерии оценки.

Выполнено одно задание, на вопросы преподавателя по заданию даны положительные ответы – оценка удовлетворительно.

Выполнены два задания, на вопросы преподавателя по заданиям даны положительные ответы – оценка хорошо.

Выполнены три задания, на вопросы преподавателя по заданиям даны положительные ответы – оценка отлично.

Задание 1. Постройте таблицу значений функции y=f(x) для

Границы диапазона и шаг изменения аргумента ввести с клавиатуры.

Задания под четным вариантом необходимо выполнить, используя цикл с предусловием, а нечетные варианты, используя цикл с постусловием.

y=

double y = 0;

Console.WriteLine("Введите начало отрезка А:");

double otr\_A = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите начало отрезка B:");

double otr\_B = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите шаг:");

int step = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Рассчет y циклом do while в табличном виде:");

do

{

if (otr\_A == 1 || otr\_A == -1)

{

y = 1;

Console.WriteLine($"x = {otr\_A,2}|y = {y,2:f2}");

otr\_A += step;

}

if (otr\_A >= 0 && otr\_A != 1)

{

y = -1 / (1 - otr\_A);

Console.WriteLine($"x = {otr\_A,2}|y = {y,2:f2}");

otr\_A += step;

}

if (otr\_A < 0 && otr\_A != 1)

{

y = 1 / (1 + otr\_A);

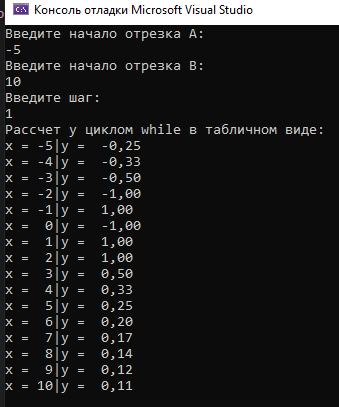
Console.WriteLine($"x = {otr\_A,2}|y = {y,2:f2}");

otr\_A += step;

}

} while (otr\_A <= otr\_B);

Console.ReadKey();



Задание 2

1. Натуральное число из n цифр является числом Армстронга, если сумма его цифр, возведенных в n-ю степень, равна самому числу (например, 153=13+53+33). 2. Получить все числа Армстронга, состоящие из трех цифр.

int one, dec, hun, m, i = 100;

do

{

m = i;

one = i % 10;

dec = (i / 10) % 10;

hun = (i / 100) % 10;

if ((Math.Pow(one, 3) + Math.Pow(dec, 3) + Math.Pow(hun, 3) == i))

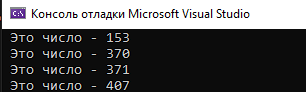
{

Console.WriteLine($" Это число - {i}");

}

i++;

} while (i < 1000);



1. Дано число A (> 1). Вывести наименьшее из целых чисел K, для которых сумма 1 + 1/2 + … + 1/K будет больше A, и саму эту сумму.

do

{

Console.WriteLine("Введите число:");

double a = Convert.ToDouble(Console.ReadLine());

int k = 0;

double tmp = 0;

do

{

k++;

tmp = tmp + 1 / (double)k;

} while (tmp <= a);

Console.WriteLine($"Наименьшее целое число K: {k}");

Console.WriteLine("Завершить работу? (Да, Нет)");

string end = Console.ReadLine();

if (end.Equals("Да") == true)

{

break;

}

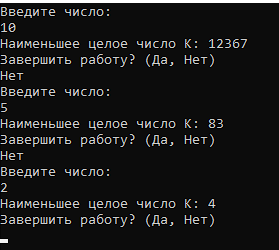
else

{

continue;

}

} while (true);

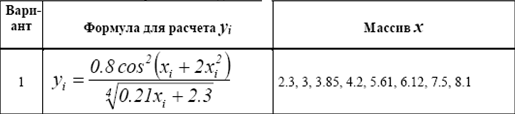


Лабораторная работа №7

Тема: Одномерные массивы.

Цель: Научиться реализовывать алгоритмы для работы с одномерными массивами средствами Visual Studio, используя консольное приложение.

Задание 1. Составить программу для решения задачи: Задан массив X. Рассчитать значения элементов массива Y согласно заданному варианту (см. таблицу 1). Вывести на экран массивы X и Y (в столбец). Найти максимальный и минимальный элементы массива Y, а также вычислить среднее арифметическое элементов массива Y.



double[] x = {2.3, 3, 3.85, 4.2, 5.61, 6.12, 7.5, 8.1};

double[] y = new double[8];

double y\_i;

double summ\_y = 0;

for (int i = 0; i < 8; i++)

{

y\_i = 0.8 \* Math.Pow(Math.Cos(x[i] + 2 \* Math.Pow(x[i], 2)), 2) / Math.Pow(0.21 \* x[i] + 2.3, 1/4);

Console.WriteLine($"|x = {x[i],2:f1}|y = {y\_i,2:f2}|");

y.SetValue(y\_i, i); // запись в массив

}

foreach (int a in y)

{

summ\_y += y[a]; // сумма элементов массива y

}

Array.Sort(y); // сортировка массива

double maxValue\_y = y[^1]; //максимальное значение

Console.WriteLine($"Максимальное значение массива y: {maxValue\_y:f2}");

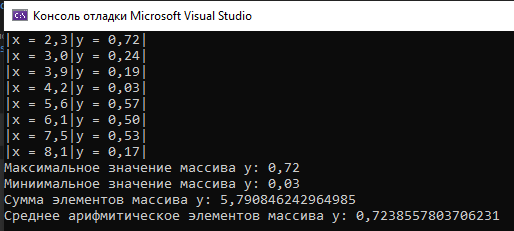
double minValue\_y = y[0]; //миниимальное значение

Console.WriteLine($"Миниимальное значение массива y: {minValue\_y:f2}");

Console.WriteLine($"Сумма элементов массива y: {summ\_y}");

double sr\_ar = summ\_y / 8;

Console.WriteLine($"Среднее арифмитическое элементов массива y: {sr\_ar}");



Задание 2. Задания выполнить с применением целочисленного одномерного массива. Размерность массива вводить с клавиатуры. Способ заполнения массива выбрать самостоятельно.

1) заменить все положительные элементы противоположными им числами;

Console.WriteLine("Введите размерность массива(число):");

int n = Convert.ToInt32(Console.ReadLine());

int[] A = Enumerable.Range(0, n).ToArray();

foreach (int i in A)

{

Console.WriteLine($"|Индекс {i,2} | Значение {A[i],2}|");

}

Console.WriteLine($"Противоположные значения элементов массива:");

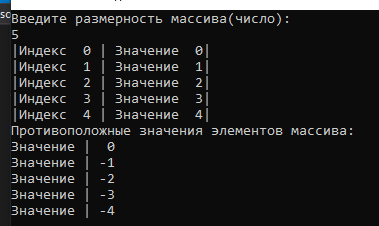
foreach (int i in A)

{

A[i] = -A[i];

Console.WriteLine($"Значение | {A[i], 2}");

}



2) заменить все элементы, меньшие заданного числа, этим числом;

Console.WriteLine("Введите размерность массива(число):");

int n = Convert.ToInt32(Console.ReadLine());

int[] A = Enumerable.Range(0, n).ToArray();

foreach (int i in A)

{

Console.WriteLine($"|Индекс {i,2} | Значение {A[i],2}|");

}

Console.WriteLine("Введите число:");

int ch = Convert.ToInt32(Console.ReadLine());

Console.WriteLine($"Элементы, меньшие заданного числа, заменённые этим числом:");

foreach (int i in A)

{

if (A[i] < ch)

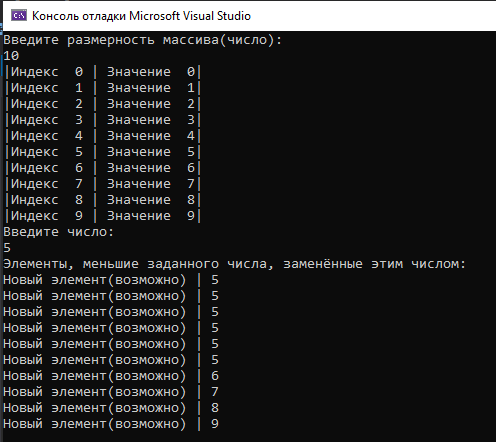
{

A[i] = ch;

}

Console.WriteLine($"Новый элемент(возможно) | {A[i]}");

}



1. заменить все элементы, попадающие в интервал [a, b], нулем;

int[] A;

int m = 0;

Console.Write("Введите число a - ");

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

Console.Write("Введите число b - ");

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

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

if (elem >= a && elem <= b)

{

Console.Write($"{m} ");

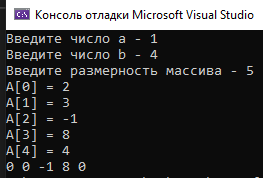
}

else

{

Console.Write($"{elem} ");

}



1. заменить все отрицательные элементы, не кратные 3, противоположными им числами;

int[] A;

int y;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

if (elem < 0 && elem % 3 != 0)

{

y = elem \* -1;

Console.Write($"{y} ");

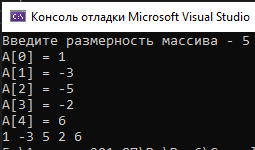
}

else

{

Console.Write($"{elem} ");

}



1. все элементы, меньшие заданного числа, увеличить в два раза;

int[] A;

int y;

Console.Write("Введите число - ");

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

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

if (elem < ch)

{

y = elem \* 2;

Console.Write($"{y} ");

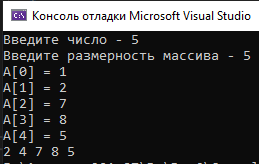
}

else

{

Console.Write($"{elem} ");

}



1. подсчитать среднее арифметическое элементов;

int[] A;

double y = 0;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

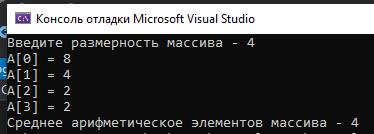
{

y += elem;

}

double sr = y / n;

Console.Write($"Среднее арифметическое элементов массива - {sr:f1} ");



1. подсчитать среднее арифметическое отрицательных элементов;

int[] A;

int z = 0;

double y = 0;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

if (elem < 0)

{

y += elem;

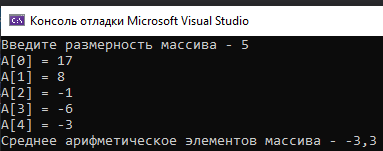
z++;

}

}

double sr = y / z;

Console.Write($"Среднее арифметическое элементов массива - {sr:f1} ");



1. подсчитать количество нечетных элементов;

int[] A;

int z = 0;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

if (elem % 2 != 0)

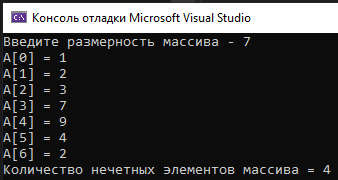
{

z++;

}

}

Console.Write($"Количество нечетных элементов массива = {z} ");



1. подсчитать сумму элементов, попадающих в заданный интервал;

int[] A;

int sum = 0;

Console.Write("Введите число a - ");

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

Console.Write("Введите число b - ");

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

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

if (elem >= a && elem <= b)

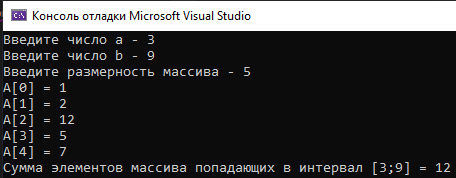
{

sum += elem;

}

}

Console.Write($"Сумма элементов массива попадающих в интервал [{a};{b}] = {sum} ");



1. подсчитать сумму элементов, кратных 9;

int[] A;

int sum = 0;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

if (elem % 9 == 0)

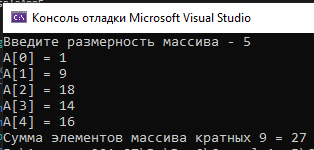
{

sum += elem;

}

}

Console.Write($"Сумма элементов массива кратных 9 = {sum} ");



1. подсчитать количество элементов, не попадающих в заданный интервал;

int[] A;

int sum = 0;

Console.Write("Введите число a - ");

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

Console.Write("Введите число b - ");

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

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

if (elem <= a || elem >= b)

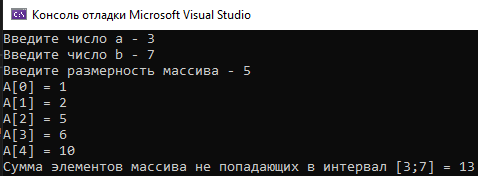
{

sum += elem;

}

}

Console.Write($"Сумма элементов массива не попадающих в интервал [{a};{b}] = {sum} ");



1. подсчитать сумму квадратов четных элементов;

int[] A;

int sum = 0;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

if (elem % 2 == 0)

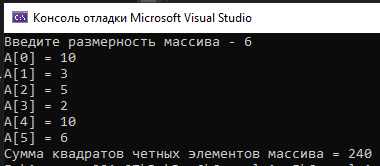
{

sum += elem\*elem;

}

}

Console.Write($"Сумма квадратов четных элементов массива = {sum} ");



1. вывести на экран номера всех элементов, больших заданного числа;

int[] A;

int u = 0;

Console.Write("Введите число a - ");

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

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

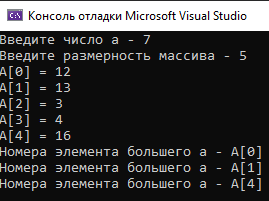
if (elem > a)

{

Console.WriteLine($"Номера элемента большего a - A[{u}] ");

}

u++;



1. вывести на экран номера всех нечетных элементов;

int[] A;

int u = 0;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

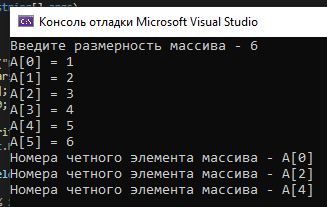
if (elem % 2 != 0)

{

Console.WriteLine($"Номера четного элемента массива - A[{u}] ");

}

u++;



1. вывести на экран номера всех элементов, которые не делятся на 7;

int[] A;

int u = 0;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

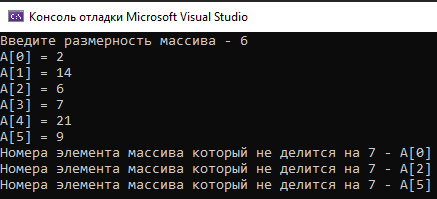
if (elem % 7 != 0)

{

Console.WriteLine($"Номера элемента массива который не делится на 7 - A[{u}] ");

}

u++;



1. вывести на экран номера всех элементов, не попадающих в заданный интервал;

int[] A;

int u = 0;

Console.Write("Введите число a - ");

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

Console.Write("Введите число b - ");

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

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

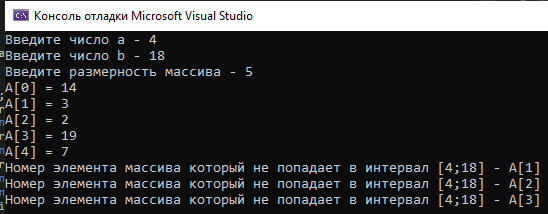
if (elem < a || elem > b)

{

Console.WriteLine($"Номер элемента массива который не попадает в интервал [{a};{b}] - A[{u}] ");

}

u++;



1. определить, является ли произведение элементов трехзначным числом;

int[] A;

int proz = 1;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

proz \*= elem;

}

if (proz >= 100 && proz < 1000)

{

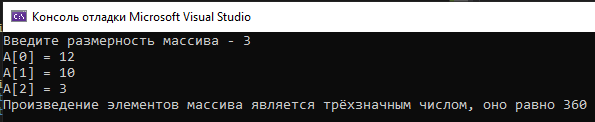
Console.WriteLine($"Произведение элементов массива является трёхзначным числом, оно равно {proz}");

}

else

{

Console.WriteLine($"Произведение элементов массива не является трёхзначным числом, оно равно {proz}");



1. определить, является ли сумма элементов двухзначным числом;

int[] A;

int sum = 0;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

sum += elem;

}

if (sum >= 10 && sum < 100)

{

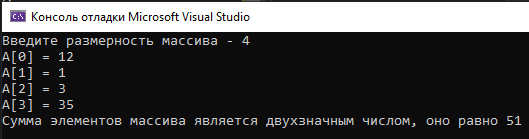
Console.WriteLine($"Сумма элементов массива является двухзначным числом, оно равно {sum}");

}

else

{

Console.WriteLine($"Сумма элементов массива не является двухзначным числом, оно равно {sum}");



1. вывести на экран элементы с четными индексами (для двумерного массива сумма индексов должна быть четной);

int[] A;

int u = 0;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

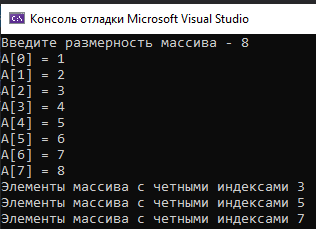
{

u++;

if (u % 2 != 0 && u > 1)

{

Console.WriteLine($"Элементы массива с четными индексами {elem}");



1. вывести на экран положительные элементы с нечетными индексами (для двумерного массива первый индекс должен быть нечетным).

int[] A;

int u = 0;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

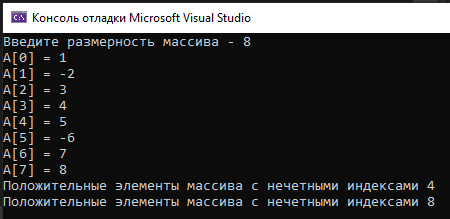
{

u++;

if (u % 2 == 0 && u != 0 && elem > 0)

{

Console.WriteLine($"Положительные элементы массива с нечетными индексами {elem}");



Задание 3.Задания выполнить с применением целочисленного одномерного массива. Размерность массива вводить с клавиатуры. Способ заполнения массива выбрать самостоятельно.

1. подсчитать количество максимальных элементов;

int[] A;

Random rnd = new Random();

Console.Write("Введите размерность массива:");

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

Console.Write("Введите верхнюю границу заполнения: ");

int n\_1 = int.Parse(Console.ReadLine());

Console.Write("Введите нижнюю границу заполнения: ");

int n\_2 = int.Parse(Console.ReadLine());

A = new int[n];

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

{

if (n\_1 < n\_2)

{

A[i] = rnd.Next(n\_1, n\_2);

}

else

{

A[i] = rnd.Next(n\_2, n\_1);

}

Console.Write($"{i} элемент массива = {A[i]} \n");

}

Array.Sort(A);

int max = A[0];

int maxi = 0;

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

{

if (A[k] > max)

{

max = A[k];

maxi = k;

}

}

int l = 0;

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

{

if (A[i] == max)

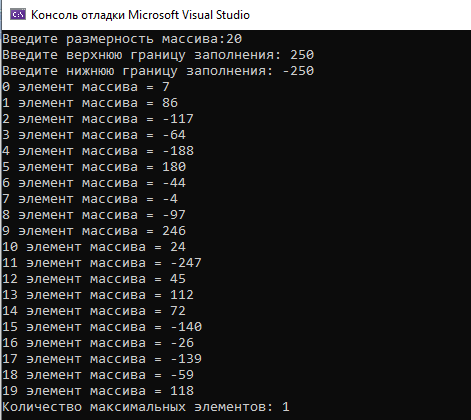
{

l++;

}

}

Console.WriteLine($"Количество максимальных элементов: {l}");



1. вывести на экран номера всех минимальных элементов;

int[] A;

Random rnd = new Random();

Console.Write("Введите размерность массива:");

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

Console.Write("Введите верхнюю границу заполнения: ");

int n\_1 = int.Parse(Console.ReadLine());

Console.Write("Введите нижнюю границу заполнения: ");

int n\_2 = int.Parse(Console.ReadLine());

A = new int[n];

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

{

if (n\_1 < n\_2)

{

A[i] = rnd.Next(n\_1, n\_2);

}

else

{

A[i] = rnd.Next(n\_2, n\_1);

}

Console.Write($"{i} элемент массива = {A[i]} \n");

}

Array.Sort(A);

int min = A[0];

int mini = 0;

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

{

if (A[k] < min)

{

min = A[k];

mini = k;

}

}

int l = 0;

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

{

if (A[i] == min)

{

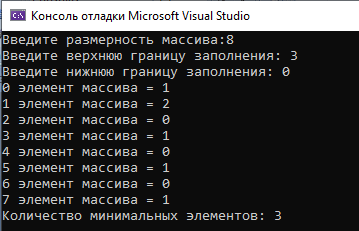
l++;

}

}

Console.WriteLine($"Количество минимальных элементов: {l}");

}



1. заменить все максимальные элементы нулями;

int[] A;

Random rnd = new Random();

Console.Write("Введите размерность массива:");

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

Console.Write("Введите верхнюю границу заполнения: ");

int n\_1 = int.Parse(Console.ReadLine());

Console.Write("Введите нижнюю границу заполнения: ");

int n\_2 = int.Parse(Console.ReadLine());

A = new int[n];

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

{

if (n\_1 < n\_2)

{

A[i] = rnd.Next(n\_1, n\_2);

}

else

{

A[i] = rnd.Next(n\_2, n\_1);

}

Console.Write($"{i} элемент массива = {A[i]} \n");

}

Array.Sort(A);

int max = A[0];

int maxi = 0;

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

{

if (A[k] > max)

{

max = A[k];

maxi = k;

}

}

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

{

if (A[i] == max)

{

A[i] = 0;

}

}

Console.WriteLine("----------------------------------------");

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

{

Console.Write($"{i} элемент массива = {A[i]} \n");

}

1. заменить все минимальные элементы на противоположные;

int[] A;

Random rnd = new Random();

Console.Write("Введите размерность массива:");

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

Console.Write("Введите верхнюю границу заполнения: ");

int n\_1 = int.Parse(Console.ReadLine());

Console.Write("Введите нижнюю границу заполнения: ");

int n\_2 = int.Parse(Console.ReadLine());

A = new int[n];

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

{

if (n\_1 < n\_2)

{

A[i] = rnd.Next(n\_1, n\_2);

}

else

{

A[i] = rnd.Next(n\_2, n\_1);

}

Console.Write($"{i} элемент массива = {A[i]} \n");

}

Array.Sort(A);

int min = A[0];

int mini = 0;

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

{

if (A[k] < min)

{

min = A[k];

mini = k;

}

}

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

{

if (A[i] == min)

{

A[i] = A[i] \* -1;

}

}

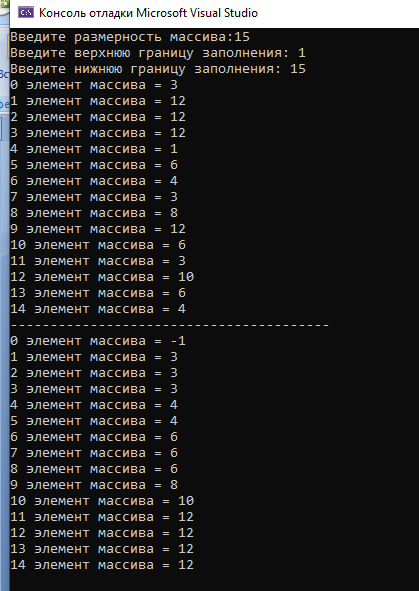
Console.WriteLine("----------------------------------------");

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

{

Console.Write($"{i} элемент массива = {A[i]} \n");

}



1. поменять местами максимальный элемент и первый;

int[] A;

Random rnd = new Random();

Console.Write("Введите размерность массива:");

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

Console.Write("Введите верхнюю границу заполнения: ");

int n\_1 = int.Parse(Console.ReadLine());

Console.Write("Введите нижнюю границу заполнения: ");

int n\_2 = int.Parse(Console.ReadLine());

A = new int[n];

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

{

if (n\_1 < n\_2)

{

A[i] = rnd.Next(n\_1, n\_2);

}

else

{

A[i] = rnd.Next(n\_2, n\_1);

}

Console.Write($"{i} элемент массива = {A[i]} \n");

}

Array.Sort(A);

int max = A[0];

int maxi = 0;

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

{

if (A[k] > max)

{

max = A[k];

maxi = k;

}

}

int tmp = 0;

int first = A[0];

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

{

if (A[i] == max)

{

tmp = first;

first = A[i];

A[i] = tmp;

}

}

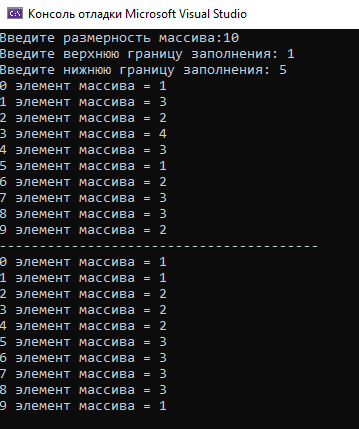
Console.WriteLine("----------------------------------------");

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

{

Console.Write($"{i} элемент массива = {A[i]} \n");

}



1. вывести на экран номера всех элементов, не совпадающих с максимальным;

using System;

namespace ConsoleApp5

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random();

int[] A;

Console.Write("Введите размерность массива - ");

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

Console.Write("Введите нижнюю границу массива - ");

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

Console.Write("Введите верхнюю границу массива - ");

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

A = new int[n];

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

{

A[i] = rnd.Next(a, b);

Console.Write($"{A[i]} ");

}

int max = A[0];

int maxi = 0;

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

{

if (A[i] > max)

{

max = A[i];

maxi = i;

}

}

Console.WriteLine(" ");

Console.Write($"Номера элементов, не совпадающих с максимальным: ");

int k = 0;

int y = A[0];

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

{

if (A[i] != max)

{

Console.Write($"{k} ");

}

k++;

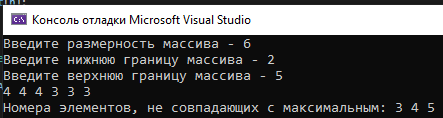
}

Console.ReadKey();

}

}

}



1. найти номер первого минимального элемента;

using System;

namespace ConsoleApp5

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random();

int[] A;

Console.Write("Введите размерность массива - ");

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

Console.Write("Введите нижнюю границу массива - ");

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

Console.Write("Введите верхнюю границу массива - ");

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

A = new int[n];

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

{

A[i] = rnd.Next(a, b);

Console.Write($"{A[i]} ");

}

int min = A[0];

int maxi = 0;

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

{

if (A[i] < min)

{

min = A[i];

maxi = i;

}

}

Console.WriteLine(" ");

Console.Write($"Номер первого минимального элемента: ");

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

{

if (A[i] < min)

{

min = A[i];

break;

}

}

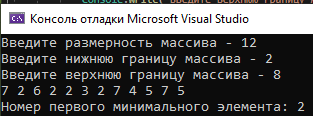
Console.Write($"{min} ");

Console.ReadKey();

}

}

}



1. найти номер последнего максимального элемента;

using System;

namespace ConsoleApp5

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random();

int[] A;

Console.Write("Введите размерность массива - ");

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

Console.Write("Введите нижнюю границу массива - ");

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

Console.Write("Введите верхнюю границу массива - ");

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

A = new int[n];

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

{

A[i] = rnd.Next(a, b);

Console.Write($"{A[i]} ");

}

int max = A[0];

int maxi = 0;

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

{

if (A[i] > max)

{

max = A[i];

maxi = i;

}

}

Console.WriteLine(" ");

Console.Write($"Номер последнего максимального элементаа: ");

int k = 0;

int t = 0;

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

{

if (A[i] >= max)

{

t = k;

}

k++;

}

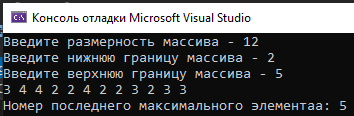
Console.Write($"{t} ");

Console.ReadKey();

}

}

}



1. подсчитать сумму элементов, расположенных между максимальным и минимальным элементами (минимальный и максимальный элементы в массиве единственные); если максимальный элемент встречается позже минимального, то выдать сообщение об этом;

using System;

namespace ConsoleApp5

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random();

int[] A;

int sum = 0;

Console.Write("Введите размерность массива - ");

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

Console.Write("Введите нижнюю границу массива - ");

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

Console.Write("Введите верхнюю границу массива - ");

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

A = new int[n];

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

{

A[i] = rnd.Next(a, b);

Console.Write($"{A[i]} ");

}

int max = A[0];

int maxi = 0;

int min = A[0];

int mini = 0;

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

{

if (A[i] > max)

{

max = A[i];

maxi = i;

}

if (A[i] < min)

{

min = A[i];

mini = i;

}

}

if (maxi < mini)

{

Console.WriteLine("\nОшибка, максимальный элемент встречается после минимального");

return;

}

for (int i = mini + 1; i < maxi; ++i)

{

sum += A[i];

}

Console.WriteLine(" ");

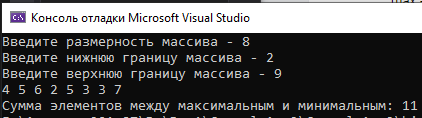
Console.Write($"Сумма элементов между максимальным и минимальным: {sum}");

Console.ReadKey();

}

}

}



1. найти номер первого максимального элемента;

using System;

namespace ConsoleApp5

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random();

int[] A;

Console.Write("Введите размерность массива - ");

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

Console.Write("Введите нижнюю границу массива - ");

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

Console.Write("Введите верхнюю границу массива - ");

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

A = new int[n];

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

{

A[i] = rnd.Next(a, b);

Console.Write($"{A[i]} ");

}

int max = A[0];

int maxi = 0;

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

{

if (A[i] > max)

{

max = A[i];

maxi = i;

}

}

Console.WriteLine(" ");

Console.Write($"Номер первого максимального элемента: ");

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

{

if (A[i] > max)

{

max = A[i];

maxi = i;

break;

}

}

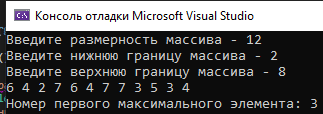
Console.Write($"{maxi} ");

Console.ReadKey();

}

}

}



1. найти номер последнего минимального элемента;

using System;

namespace ConsoleApp5

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random();

int[] A;

Console.Write("Введите размерность массива - ");

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

Console.Write("Введите нижнюю границу массива - ");

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

Console.Write("Введите верхнюю границу массива - ");

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

A = new int[n];

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

{

A[i] = rnd.Next(a, b);

Console.Write($"{A[i]} ");

}

int min = A[0];

int maxi = 0;

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

{

if (A[i] < min)

{

min = A[i];

maxi = i;

}

}

Console.WriteLine(" ");

Console.Write($"Номер последнего минимального элемента: ");

int k = 0;

int t = 0;

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

{

if (A[i] == min)

{

t = k;

}

k++;

}

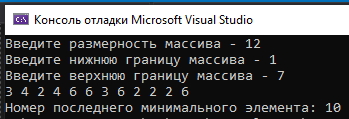
Console.Write($"{t} ");

Console.ReadKey();

}

}

}



1. подсчитать сумму элементов, расположенных между первым максимальным и последним минимальными элементами; если максимальный элемент встречается позже минимального, то выдать сообщение об этом;

using System;

namespace ConsoleApp5

{

class Program

{

static void Main(string[] args)

{

int[] A;

int sum = 0;

Console.Write("Введите количество элементов массива: ");

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

A = new int[n];

Console.WriteLine("Введите левую границу");

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

Console.WriteLine("Введите правую границу");

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

Random rand = new Random();

Console.Write("Массив: ");

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

{

A[i] = rand.Next(C, D);

Console.Write($"{A[i]} ");

}

int max = A[0];

int maxi = 0;

int min = A[0];

int mini = 0;

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

{

if (A[i] < min)

{

min = A[i];

mini = i;

}

if (A[i] > max)

{

max = A[i];

maxi = i;

}

}

if (maxi < mini)

{

Console.WriteLine("\nОшибка, минимального элемент встречается после максимальный");

return;

}

for (int i = mini + 1; i < maxi; ++i)

{

sum += A[i];

}

Console.WriteLine(" ");

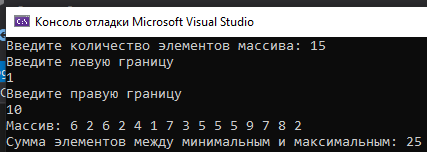
Console.Write($"Сумма элементов между минимальным и максимальным: {sum}");

Console.ReadKey();

}

}

}



1. поменять местами первый минимальный и последний максимальный элементы;

using System;

namespace ConsoleApp5

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random();

int[] A;

Console.Write("Введите размерность массива - ");

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

Console.Write("Введите нижнюю границу массива - ");

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

Console.Write("Введите верхнюю границу массива - ");

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

A = new int[n];

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

{

A[i] = rnd.Next(a, b);

Console.Write($"{A[i]} ");

}

int min = A[0];

int mini = 0;

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

{

if (A[i] < min)

{

min = A[i];

mini = i;

}

}

Console.WriteLine(" ");

Console.Write($"Первый минимальный элемент: ");

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

{

if (A[i] < min)

{

min = A[i];

break;

}

}

Console.Write($"{min} ");

int max = A[0];

int maxi = 0;

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

{

if (A[i] < max)

{

max = A[i];

maxi = i;

}

}

Console.WriteLine(" ");

Console.Write($"Последний максимальный элемент: ");

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

{

if (A[i] > max)

{

max = A[i];

}

}

Console.Write($"{max} ");

int y = min;

int x = max;

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

{

if (A[i] == min)

{

A[i] = min;

}

if (A[i] == max)

{

A[i] = max;

}

}

Console.WriteLine(" ");

Console.Write($"Новый массив: ");

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

{

Console.Write($"{A[i]} ");

}

Console.ReadKey();

}

}

}

1. найти максимум из отрицательных элементов;

using System;

namespace ConsoleApp5

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random();

int[] A;

Console.Write("Введите размерность массива - ");

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

Console.Write("Введите нижнюю границу массива - ");

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

Console.Write("Введите верхнюю границу массива - ");

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

A = new int[n];

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

{

A[i] = rnd.Next(a, b);

Console.Write($"{A[i]} ");

}

int max = A[0];

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

{

if ( max < 0 && A[i] < 0 && A[i] > max)

{

max = A[i];

}

}

Console.WriteLine(" ");

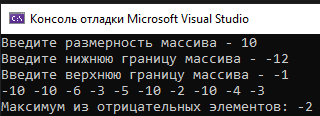
Console.Write($"Максимум из отрицательных элементов: {max}");

Console.ReadKey();

}

}

}



1. найти минимум из положительных элементов;

using System;

namespace ConsoleApp5

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random();

int[] A;

Console.Write("Введите размерность массива - ");

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

Console.Write("Введите нижнюю границу массива - ");

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

Console.Write("Введите верхнюю границу массива - ");

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

A = new int[n];

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

{

A[i] = rnd.Next(a, b);

Console.Write($"{A[i]} ");

}

int min = A[0];

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

{

if (A[i] > 0 && min > 0 && A[i] < min)

{

min = A[i];

}

}

Console.WriteLine(" ");

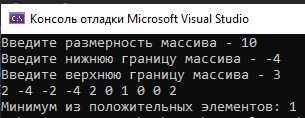
Console.Write($"Минимум из положительных элементов: {min}");

Console.ReadKey();

}

}

}



1. найти максимум из модулей элементов;

using System;

namespace ConsoleApp5

{

class Program

{

static void Main(string[] args)

{

Random rnd = new Random();

int[] A;

Console.Write("Введите размерность массива - ");

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

Console.Write("Введите нижнюю границу массива - ");

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

Console.Write("Введите верхнюю границу массива - ");

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

A = new int[n];

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

{

A[i] = rnd.Next(a, b);

Console.Write($"{A[i]} ");

}

int max = A[0];

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

{

if (Math.Abs(A[i]) > Math.Abs(max))

{

max = A[i];

}

}

Console.WriteLine(" ");

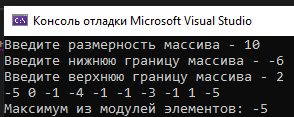
Console.Write($"Максимум из модулей элементов: {max}");

Console.ReadKey();

}

}

}



Контрольная работа (Одномерные массивы) №5

11. Дан массив целых чисел. Размерность массива указать с клавиатуры. Элементы массива вводить с клавиатуры. Найти все элементы, кратные 3 или 5.

int[] A;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

foreach (int elem in A)

{

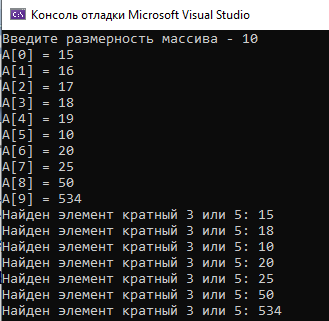
if (elem % 5 == 0 || elem % 3 == 0)

{

Console.WriteLine($"Найден элемент кратный 3 или 5: {elem}");

}

}



16. Дан массив 15 целых чисел. Элементы массива заполняются случайно с помощью генератора случайных чисел из диапазона [15,35]. Найти сумму элементов, имеющих нечетное значение.

Random rnd = new Random();

int[] A;

int n = 15;

int a = 15;

int b = 36;

A = new int[n];

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

{

A[i] = rnd.Next(a, b);

Console.WriteLine($"A[{i}] = {A[i]}");

}

int sum = 0;

foreach (int elem in A)

{

if (elem % 2 == 1)

{

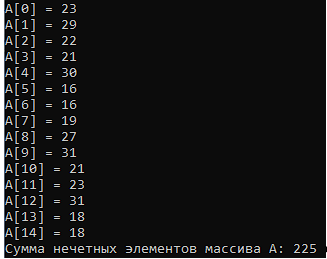
sum += elem;

}

}

Console.WriteLine($"Сумма нечетных элементов массива А: {sum}");

Console.ReadKey();



21. Дан массив 20 целых чисел. Элементы массива вводить с клавиатуры. Найти сумму отрицательных элементов.

int[] A;

int n = 20;

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

int sum = 0;

foreach (int elem in A)

{

if (elem < 0)

{

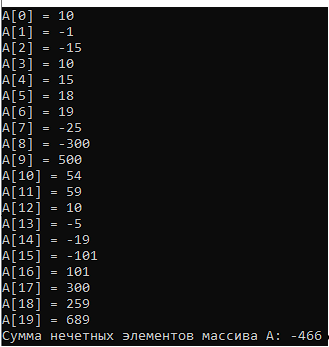
sum += elem;

}

}

Console.WriteLine($"Сумма нечетных элементов массива А: {sum}");

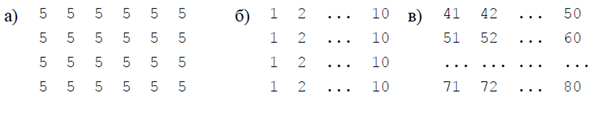
Console.ReadKey();



Лабораторная работа 8

Тема: Циклы (Вложенные циклы)

Напечатать числа в виде следующей таблицы:



А)

int i = 0;

int j = 0;

for (i = 0; i < 4; i++)

{

for (j = 0; j < 5; j++)

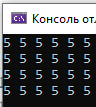
{

Console.Write("5 ");

}

Console.WriteLine("5 ");

}



Б)

for (i = 0; i < 4; i++)

{

for (j = 0; j < 11; j++)

{

if (j > 0)

{

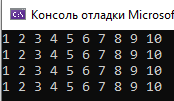
Console.Write($"{j} ");

}

}

Console.WriteLine("");

}



В)

int i = 0;

int j = 0;

int a = 41;

int b = 50;

for (i = 0; i < 4; i++)

{

for (j = a; j <= b; j++)

{

Console.Write($"{j} ");

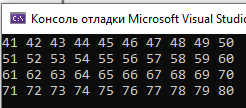
}

a += 10;

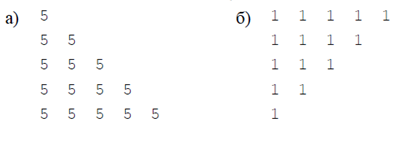
b += 10;

Console.WriteLine("");

}



Напечатать числа в виде следующей таблицы:



А)

int i = 0;

int j = 0;

int a = 2;

for (i = 0; i < 5; i++)

{

for (j = 1; j < a; j++)

{

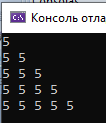
Console.Write($"5 ");

}

a++;

Console.WriteLine("");

}



Б)

int i = 0;

int j = 0;

int a = 0;

for (i = 0; i < 5; i++)

{

for (j = 5; j > a; j--)

{

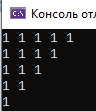
Console.Write($"1 ");

}

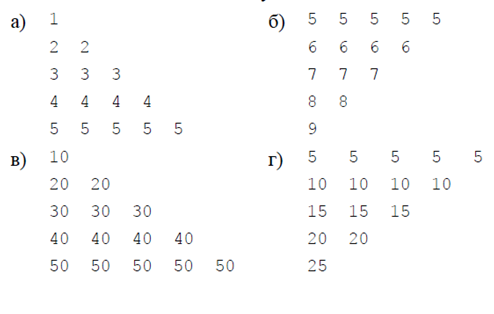
a++;

Console.WriteLine("");

}



Напечатать числа в виде следующей таблицы:



А)

int i = 0;

int j = 0;

int a = 2;

int b = 1;

for (i = 0; i < 5; i++)

{

for (j = 1; j < a; j++)

{

Console.Write($"{b} ");

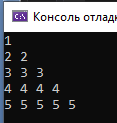
}

a++;

b++;

Console.WriteLine("");

}



Б)

int i = 0;

int j = 0;

int a = 0;

int b = 5;

for (i = 0; i < 5; i++)

{

for (j = 5; j > a; j--)

{

Console.Write($"{b} ");

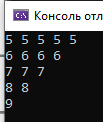
}

a++;

b++;

Console.WriteLine("");

}



В)

int i = 0;

int j = 0;

int a = 2;

int b = 10;

for (i = 0; i < 5; i++)

{

for (j = 1; j < a; j++)

{

Console.Write($"{b} ");

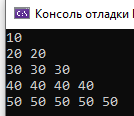
}

a++;

b += 10;

Console.WriteLine("");

}



Г)

int i = 0;

int j = 0;

int a = 0;

int b = 5;

for (i = 0; i < 5; i++)

{

for (j = 5; j > a; j--)

{

Console.Write($"{b} ");

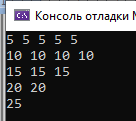
}

a++;

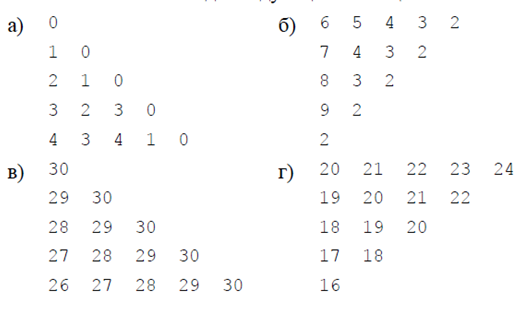
b += 5;

Console.WriteLine("");

}



Напечатать числа в виде следующей таблицы:



А)

int i;

for (i = 0; i < 5; i++)

{

int j;

for (j = 0; j <= i ; j++)

{

if (j == i)

{

Console.Write("0");

}

else

{

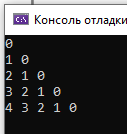
Console.Write(i - j + " ");

}

}

Console.WriteLine(" ");

}



Б)

int i;

for (i = 0; i < 5; i++)

{

int j;

for (j = 6; j > i; j--)

{

if (j > i+1)

{

if (j == i)

{

Console.Write(2);

}

else

{

Console.Write(j - i + " ");

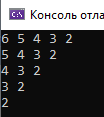
}

}

}

Console.WriteLine("");

}



В)

for (int n = 0; n < 5; n++)

{

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

{

if (i == n)

{

Console.Write(30);

}

else

{

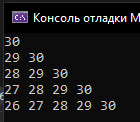
Console.Write( 30 + i - n + " ");

}

}

Console.WriteLine(" ");

}



Г)

for (int n = 20; n > 15; n--)

{

for (int i = 0; i < n - 15; i++)

{

if (i == n)

{

Console.Write(16);

}

else

{

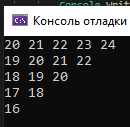
Console.Write(n + i + " ");

}

}

Console.WriteLine(" ");

}



Лабораторная работа 9

Тема.Статические методы.

Цель: научиться использовать статические методы с параметры и без параметров для выполнения консольных приложений расчетного характера.

Задание 1. Разработать консольное приложение в соответствии с условием задания.

Задания сгруппированы. Решить по одной задаче из каждой группы.

Критерии оценки задания.

Решены правильно две задачи – оценка удовлетворительно.

Решены правильно или с недочетами четыре задачи – оценка хорошо.

Решены пять или шесть задач, ответы на дополнительные вопросы правильные – оценка отлично.

Группа 1.

Разработать метод min(a,b) для нахождения минимального из двух чисел. Вычислить с его помощью значение выражения z=min(3x,2y)+min(x-y,x+y).z

static void Main(string[] args)

{

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

double a = Convert.ToDouble(Console.ReadLine());

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

double b = Convert.ToDouble(Console.ReadLine());

Console.WriteLine($"Минимальное число: {Min(a,b)}");

double z = Min(3 \* a, 2 \* b) + Min(a - b, a + b);

Console.WriteLine($"Значение выражения z: {z}");

}

static double Min(double a, double b)

{

if (a > b)

{

return b;

}

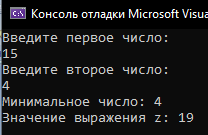
else

{

return a;

}

}



Группа 2.

Разработать метод f(x), который вычисляет значение по следующей формуле: f(x)=x3-sin (x).

Определить, в какой из точек: а или b, функция принимает наибольшее значение. Значения a и b ввести с клавиатуры.

static void Main(string[] args)

{

double maxSin = 0;

Console.WriteLine("Введите число:");

double x = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите первую точку:");

double a = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("Введите вторую точку:");

double b = Convert.ToDouble(Console.ReadLine());

Console.WriteLine($"Значение выражения: {FuncSin(x):f2}");

double f1 = FuncSin(a);

double f2 = FuncSin(b);

if (f1 > f2)

{

maxSin = a;

}

else

{

maxSin = b;

}

Console.WriteLine($"Максимальное значение функция принимает в точке: {maxSin}");

}

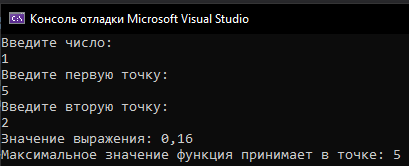
static double FuncSin(double x)

{

double f = Math.Pow(x, 3) - Math.Sin(x);

return f;

}



Группа 3.

Разработать метод f(n, x), которая для заданного натурального числа n и вещественного х находит значение выражения . Вычислить с помощью данного метода значение выражения .

static void Main(string[] args)

{

int n = 2;

double x = 2;

double expression = FuncExpr(x, n) + FuncExpr(x, 2 \* n) + FuncExpr(x, 3 \* n);

Console.WriteLine($"Значение выражения: {expression}");

}

static double FuncExpr(double x, int n)

{

double f = Math.Pow(x, n) / n;

return f;

}



Группа 4.

Разработать метод f(x), который нечетное число заменяет на 0, а четное число уменьшает в два раза. Продемонстрировать работу данного метода на примере.

static void Main(string[] args)

{

while (true)

{

Console.WriteLine("Введите число:");

double x = Convert.ToDouble(Console.ReadLine());

Console.WriteLine($"Значение выполненной функции: {FuncChetNechet(x)}");

}

}

static double FuncChetNechet(double x)

{

double f = 0;

if (x % 2 == 0)

{

f = x / 2;

}

else

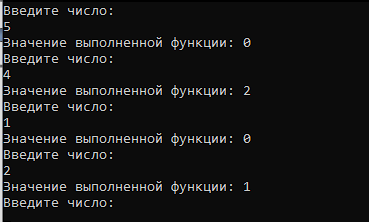
{

f = 0;

}

return f;

}



Группа 5.

Разработать метод f(a, b), который по катетам a и b вычисляет гипотенузу. С помощью данного метода найти периметр фигуры

ABDС по заданным сторонам AB, AC и DC.

using System;

namespace ConsoleApp5

{

class Program

{

static double Func1(double AB, double AC)

{

double gip = Math.Sqrt(Math.Pow(AB, 2)+ Math.Pow(AC, 2));

return gip;

}

static double Func2(double DC, double BC)

{

double gip = Math.Sqrt(Math.Pow(DC, 2) + Math.Pow(BC, 2));

return gip;

}

static void Main(string[] args)

{

Console.Write("Введите сторону AB: ");

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

Console.Write("Введите сторону AC: ");

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

Console.WriteLine($"Гипотенуза BC: {Func1(AB, AC):f2}");

Console.Write("Введите сторону DC: ");

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

double BC = Func1(AB, AC);

Console.WriteLine($"{BC}");

Console.WriteLine($"Гипотенуза BD: {Func2(DC, BC):f2}");

double per = AB + AC + DC + Func2(DC, BC);

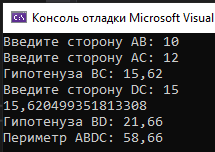
Console.WriteLine($"Периметр ABDC: {per:f2}");

Console.ReadKey();

}

}

}



Группа 6.

1. Разработать метод f(), который вычисляет длину отрезка по координатам вершин () и (), и метод d(a, b, c), который вычисляет периметр треугольника по длинам сторон a, b, c. С помощью данных методов найти периметр треугольника, заданного координатами своих вершин.

using System;

namespace ConsoleApp5

{

class Program

{

static double Func1(double x1, double x2, double y1, double y2)

{

double d = Math.Sqrt(Math.Pow(x2 - x1, 2)+ Math.Pow(y2 - y1, 2));

return d;

}

static double Func2(double a, double b, double c)

{

double per = a + b + c;

return per;

}

static void Main(string[] args)

{

Console.Write("Введите сторону x1: ");

double x1 = double.Parse(Console.ReadLine());

Console.Write("Введите сторону y1: ");

double y1 = double.Parse(Console.ReadLine());

Console.Write("Введите сторону x2: ");

double x2 = double.Parse(Console.ReadLine());

Console.Write("Введите сторону y2: ");

double y2 = double.Parse(Console.ReadLine());

Console.WriteLine($"Длина: {Func1(x1, x2, y1, y2):f2}");

Console.Write("Введите длину стороны a: ");

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

Console.Write("Введите длину стороны b: ");

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

Console.Write("Введите длину стороны c: ");

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

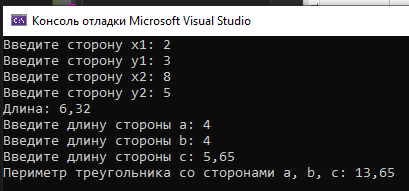
Console.WriteLine($"Периметр треугольника со сторонами a, b, c: {Func2(a, b, c):f2}");

Console.ReadKey();

}

}

}



Задание 2. Разработать консольное приложение по варианту.

1. Разработать метод, который для заданного натурального числа N возвращает значение true, если число простое и false, если число составное. С помощью данного метода:
2. вывести на экран все простые числа на отрезке [a, b];
3. найти количество всех простых чисел на отрезке [a, b];
4. найти сумму всех составных чисел на отрезке [a, b];
5. для заданного числа А вывести на экран ближайшее предшествующее по отношению к нему простое число.

using System;

namespace ConsoleApp1

{

class Program

{

static void Func(int n, ref bool t)

{

int x = n;

int k = 0;

for (int i = 1; i <= n / 2; i++)

{

if (x % i == 0)

{

k++;

}

}

if (k > 1)

{

t = false;

}

else

{

t = true;

}

}

static void Main()

{

int kol = 0;

int sum = 0;

int B = 0;

Console.Write("Введите число A: ");

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

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

Console.WriteLine($"Простые числа на отрезке [{a};{b}]");

bool t = false;

for (int i = a; i <= b; i++)

{

Func(i, ref t);

if (t == true)

{

Console.WriteLine($"{i}");

kol++;

if (A > i)

{

B = i;

}

}

else

{

sum += i;

}

t = false;

}

Console.WriteLine($"Количество простых чисел на отрезке [{a};{b}] равно {kol}");

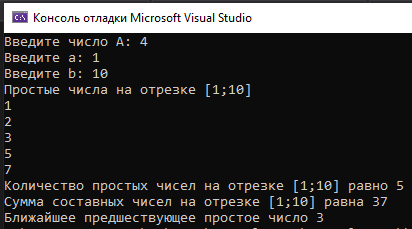
Console.WriteLine($"Сумма составных чисел на отрезке [{a};{b}] равна {sum}");

Console.Write($"Ближайшее предшествующее простое число {B}");

}

}

}



1. Разработать метод, который для заданного натурального числа N возвращает количество его делителей. С помощью данного метода:

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int del = 0;

for (int i = 1; i <= n / 2; i++)

{

if (n % i == 0)

{

del++;

}

}

del = del + 1;

return del;

}

static void Main()

{

Console.Write("Введите число N: ");

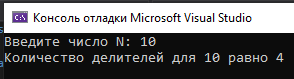
int N = int.Parse(Console.ReadLine());

Console.WriteLine($"Количество делителей для {N} равно {Func(N)}");

}

}

}



1. для каждого целого числа на отрезке [a, b] вывести на экран количество делителей;

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int del = 0;

for (int i = 1; i <= n / 2; i++)

{

if (n % i == 0)

{

del++;

}

}

del = del + 1;

return del;

}

static void Main()

{

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

for (int i = a; i <= b; i++)

{

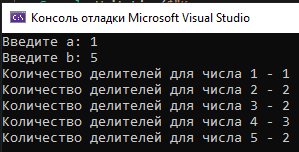
Console.WriteLine($"Количество делителей для числа {i} - {Func(i)}");

}

}

}

}



1. вывести на экран только те целые числа отрезка [a, b], у которых количество делителей равно заданному числу;

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int del = 0;

for (int i = 1; i <= n / 2; i++)

{

if (n % i == 0)

{

del++;

}

}

del = del + 1;

return del;

}

static void Main()

{

Console.Write("Введите N: ");

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

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

for (int i = a; i <= b; i++)

{

if (Func(i) == N)

{

Console.WriteLine($"Количество делителей для числа {i} - {Func(i)}");

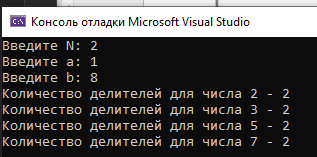
}

}

}

}

}



1. вывести на экран только те целые числа отрезка [a, b], у которых количество делителей максимально;

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int del = 0;

for (int i = 1; i <= n / 2; i++)

{

if (n % i == 0)

{

del++;

}

}

del = del + 1;

return del;

}

static void Main()

{

int max = 0;

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

for (int i = a; i <= b; i++)

{

if (Func(i) > max)

{

max = Func(i);

}

}

Console.Write($"Числа, у которых количество делителей максимально: ");

for (int i = a; i <= b; i++)

{

if (Func(i) == max)

{

Console.Write($" {i}");

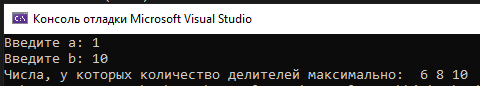
}

}

}

}

}



1. для заданного числа А вывести на экран ближайшее следующее по отношению к нему число, имеющее столько же делителей, сколько и число А.

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int del = 0;

for (int i = 1; i <= n / 2; i++)

{

if (n % i == 0)

{

del++;

}

}

del = del + 1;

return del;

}

static void Main()

{

Console.Write("Введите число A: ");

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

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

for (int i = a; i <= b; i++)

{

if(Func(i) == Func(A) && A < i)

{

Console.Write($"Ближайшее следующее число, к числу {A}: {i}");

break;

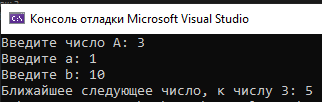
}

}

}

}

}



1. Разработать метод, который для заданного натурального числа N возвращает сумму его делителей. С помощью данного метода:

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int sum = 0;

for (int i = 1; i <= n / 2; i++)

{

if (n % i == 0)

{

sum += i;

}

}

sum = sum + n;

return sum;

}

static void Main()

{

Console.Write("Введите число N: ");

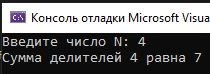
int N = int.Parse(Console.ReadLine());

Console.WriteLine($"Сумма делителей {N} равна {Func(N)}");

}

}

}



1. для каждого целого числа на отрезке [a, b] вывести на экран сумму его делителей;

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int sum = 0;

for (int i = 1; i <= n / 2; i++)

{

if (n % i == 0)

{

sum += i;

}

}

sum = sum + n;

return sum;

}

static void Main()

{

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

for (int i = a; i <= b; i++)

{

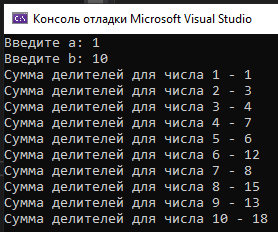
Console.WriteLine($"Сумма делителей для числа {i} - {Func(i)}");

}

}

}

}



1. вывести на экран только те целые числа отрезка [a, b], у которых сумма делителей равна заданному числу;

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int sum = 0;

for (int i = 1; i <= n / 2; i++)

{

if (n % i == 0)

{

sum += i;

}

}

sum = sum + n;

return sum;

}

static void Main()

{

Console.Write("Введите N: ");

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

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

for (int i = a; i <= b; i++)

{

if (Func(i) == N)

{

Console.WriteLine($"Сумма делителей для числа {i} - {Func(i)}");

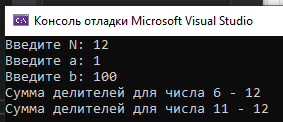
}

}

}

}

}



1. вывести на экран только те целые числа отрезка [a, b], у которых сумма делителей максимальна;

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int sum = 0;

for (int i = 1; i <= n / 2; i++)

{

if (n % i == 0)

{

sum += i;

}

}

sum = sum + n;

return sum;

}

static void Main()

{

int max = 0;

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

for (int i = a; i <= b; i++)

{

if (Func(i) > max)

{

max = Func(i);

}

}

Console.Write($"Числа, у которых сумма делителей максимально: ");

for (int i = a; i <= b; i++)

{

if (Func(i) == max)

{

Console.Write($" {i}");

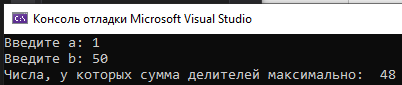
}

}

}

}

}



1. для заданного числа А вывести на экран ближайшее предшествующее по отношению к нему число, сумма делителей которого равна сумме делителей числа А.

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int sum = 0;

for (int i = 1; i <= n / 2; i++)

{

if (n % i == 0)

{

sum += i;

}

}

sum = sum + n;

return sum;

}

static void Main()

{

Console.Write("Введите число A: ");

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

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

for (int i = a; i <= b; i++)

{

if (Func(i) == Func(A) && A > i)

{

Console.Write($"Ближайшее предшествующее число, к числу {A}: {i}");

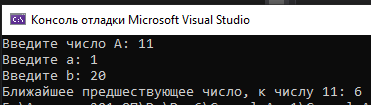
}

}

}

}

}



1. Разработать метод, которая для заданного натурального числа N возвращает сумму его цифр. С помощью данной функции:

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int sum = 0;

while(n > 0)

{

sum += n % 10;

n = n / 10;

}

return sum;

}

static void Main()

{

Console.Write("Введите число A: ");

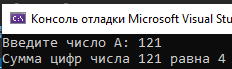
int A = int.Parse(Console.ReadLine());

Console.WriteLine($"Сумма цифр числа {A} равна {Func(A)} ");

}

}

}



1. для каждого целого числа на отрезке [a, b] вывести на экран сумму его цифр;

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int sum = 0;

while (n > 0)

{

sum += n % 10;

n = n / 10;

}

return sum;

}

static void Main()

{

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

for (int i = a; i <= b; i++)

{

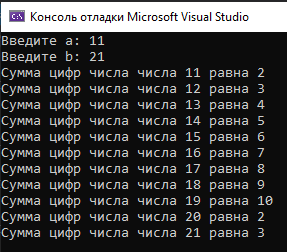
Console.WriteLine($"Сумма цифр числа числа {i} равна {Func(i)}");

}

}

}

}



1. вывести на экран только те целые числа отрезка [a, b], у которых сумма цифр числа равна заданному значению;

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int sum = 0;

while (n > 0)

{

sum += n % 10;

n = n / 10;

}

return sum;

}

static void Main()

{

Console.Write("Введите N: ");

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

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

for (int i = a; i <= b; i++)

{

if (Func(i) == N)

{

Console.WriteLine($"Сумма цифр числа {i} равна {Func(i)}");

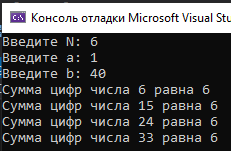
}

}

}

}

}



1. вывести на экран только те целые числа отрезка [a, b], у которых сумма цифр нечетная;

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int sum = 0;

while (n > 0)

{

sum += n % 10;

n = n / 10;

}

return sum;

}

static void Main()

{

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

for (int i = a; i <= b; i++)

{

if (Func(i) % 2 == 0)

{

Console.WriteLine($"Сумма цифр числа {i} чётная и равна {Func(i)}");

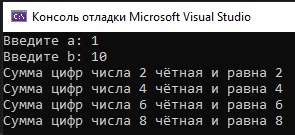
}

}

}

}

}



1. для заданного числа А вывести на экран ближайшее предшествующее по отношению к нему число, сумма цифр которого равна сумме цифр числа А.

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int n)

{

int sum = 0;

while (n > 0)

{

sum += n % 10;

n = n / 10;

}

return sum;

}

static void Main()

{

Console.Write("Введите число A: ");

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

Console.Write("Введите a: ");

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

Console.Write("Введите b: ");

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

for (int i = a; i <= b; i++)

{

if (Func(i) == Func(A) && A > i)

{

Console.Write($"Ближайшее предшествующее число, к числу {A}: {i}");

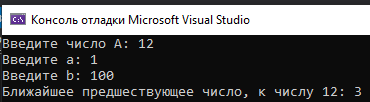
}

}

}

}

}



1. Разработать метод, которай для заданных натуральных чисел возвращает их наибольший общий делитель. С помощью данной функции:

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int a, int b)

{

int max, min;

if (Math.Abs(a) > Math.Abs(b))

{

max = Math.Abs(a);

min = Math.Abs(b);

}

Else

{

max = Math.Abs(b);

min = Math.Abs(a);

}

int r = max % min;

if (r == 0)

{

return min;

}

else

{

if (min % r == 0)

{

return r;

}

else

{

return 1;

}

}

} }

static void Main()

{

Console.Write("Введите число A: ");

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

Console.Write("Введите число B: ");

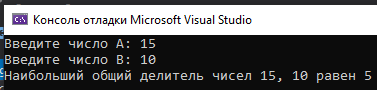
int B = int.Parse(Console.ReadLine());

Console.WriteLine($"Наибольший общий делитель чисел {A}, {B} равен {Func(A, B)}");

}

}

}



1. сократить дробь вида a/b;

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int a, int b)

{

int max, min;

if (Math.Abs(a) > Math.Abs(b))

{

max = Math.Abs(a);

min = Math.Abs(b);

}

else

{

max = Math.Abs(b);

min = Math.Abs(a);

}

int r = max % min;

if (r == 0)

{

return min;

}

else

{

if (min % r == 0)

{

return r;

}

else

{

return 1;

}

}

static void Main()

{

Console.Write("Введите числитель дроби: ");

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

Console.Write("Введите знаменатель дроби: ");

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

int ch = A / Func(A, B);

int zn = B / Func(A, B);

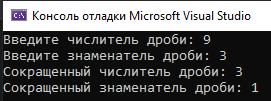
Console.WriteLine($"Сокращенный числитель дроби: {ch}");

Console.WriteLine($"Сокращенный знаменатель дроби: {zn}");

}

}

}



1. найти наименьшее общее кратное для двух натуральных чисел;

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int a, int b)

{

int max, min;

if (Math.Abs(a) > Math.Abs(b))

{

max = Math.Abs(a);

min = Math.Abs(b);

}

else

{

max = Math.Abs(b);

min = Math.Abs(a);

}

int r = max % min;

if (r == 0)

{

return min;

}

else

{

return r;

}

}

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());

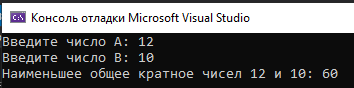
int krat = (A \* B) / Func(A, B);

Console.WriteLine($"Наименьшее общее кратное чисел {A} и {B}: {krat}");

}

}

}



1. вычислить значение выражения результат представить в виде обыкновенной дроби, выполнив сокращение;

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int a, int b)

{

int max, min;

if (Math.Abs(a) > Math.Abs(b))

{

max = Math.Abs(a);

min = Math.Abs(b);

}

else

{

max = Math.Abs(b);

min = Math.Abs(a);

}

int r = max % min;

if (r == 0)

{

return min;

}

else

{

if (min % r == 0)

{

return r;

}

else

{

return 1;

}

}

static void Main()

{

Console.Write("Введите числитель дроби: ");

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

Console.Write("Введите знаменатель дроби: ");

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

Console.Write("Введите числитель дроби: ");

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

Console.Write("Введите знаменатель дроби: ");

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

int krat = (B \* C) / Func(B, C);

int zn1 = (krat / B)\* A;

int zn2 = (krat / C) \* D;

int sum = zn1 + zn2;

Console.WriteLine($"Числитель полученной дроби: {sum}");

Console.WriteLine($"Знаменатель полученной дроби: {krat}");

int ch = sum / Func(sum, krat);

int zn = krat / Func(sum, krat);

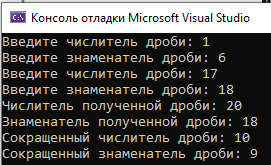
Console.WriteLine($"Сокращенный числитель дроби: {ch}");

Console.WriteLine($"Сокращенный знаменатель дроби: {zn}");

}

}

}



1. найти наибольший общий делитель для n натуральных чисел.

using System;

namespace ConsoleApp1

{

class Program

{

static int Func(int a, int b)

{

int max, min;

if (Math.Abs(a) > Math.Abs(b))

{

max = Math.Abs(a);

min = Math.Abs(b);

}

else

{

max = Math.Abs(b);

min = Math.Abs(a);

}

int r = max % min;

if (r == 0)

{

return min;

}

else

{

if (min % r == 0)

{

return r;

}

else

{

return 1;

}

}

}

static void Main()

{

int[] A;

int r = 0;

int t = 0;

Console.Write("Введите размерность массива - ");

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

A = new int[n];

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

{

Console.Write($"A[{i}] = ");

A[i] = int.Parse(Console.ReadLine());

}

int z = 1;

while ( z < n)

{

r = Func(A[0], A[z]);

t = Func(r, A[z]);

z++;

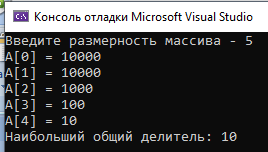
}

Console.WriteLine($"Наибольший общий делитель: {t}");

}

}

}



Контрольная работа

Тема. Статические методы

Задание 1. При выполнении задания использовать метод с типом результата отличным от void.

4. Даны действительные числа s, t. Вычислить значение y, используя метод для вычисления функции h(a,b)

y=

using System;

namespace ConsoleApp2

{

class Program

{

static double h(double a, double b)

{

double h = (a / (1 + Math.Pow(b, 2)) + (b / (1 + Math.Pow(a, 2))) - Math.Pow((a - b), 3));

return h;

}

static void Main(string[] args)

{

Console.Write("Введите число s: ");

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

Console.Write("Введите число t: ");

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

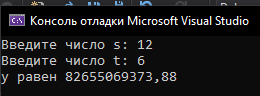
double y = h(s, t) + Math.Pow(h(s-t, s\*t), 2)+ h(1, 1);

Console.WriteLine($"y равен {y:f2}");

}

}

}



Задание 2. При выполнении задания использовать метод с типом результата void.

4. Разработать метод TrianglePS(), вычисляющий по стороне a равностороннего треугольника его периметр P = 3·a и площадь

С помощью метода TrianglePS() найти в главном методе периметры и площади трех равносторонних треугольников с данными сторонами.

using System;

namespace ConsoleApp2

{

class Program

{

static void TrianglePS(double a, ref double P, ref double S)

{

P = a \* 3;

S = (Math.Pow(a, 2) \* Math.Sqrt(3)) / 4;

}

static void Main(string[] args)

{

for (int i = 0; i < 3; i++)

{

Console.Write("Введите сторону треугольника a: ");

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

double S = 0;

double P = 0;

TrianglePS(a, ref P, ref S);

Console.WriteLine($"Периметр треугольника равен {P} ");

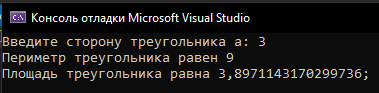
Console.WriteLine($"Площадь треугольника равна {S}; ");

}

}

}

}



Задание 3. При выполнении задания использовать метод с типом результата отличным от void.

4. Разработать метод - функцию RingS(R1, R2) вещественного типа, находящую площадь кольца, заключенного между двумя окружностями с общим центром и радиусами R1 и R2 (R1 и R2 — вещественные, R1 > R2). В главном методе с ее помощью найти площади трех колец, для которых даны внешние и внутренние радиусы. Воспользоваться формулой площади круга радиуса R: S = π·R2. В качестве значения π использовать 3.14.

using System;

namespace ConsoleApp2

{

class Program

{

static double RingS(double R1, double R2)

{

return 3.14 \* (R1 + R2) \* (R1 - R2);

}

static void Main(string[] args)

{

for (int i = 0; i < 3; i++)

{

Console.Write("Введите внешний радиус кольца: ");

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

Console.Write("Введите внутрений радиус кольца: ");

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

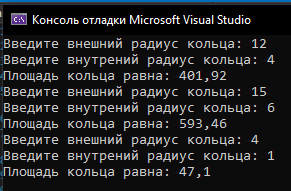
Console.WriteLine($"Площадь кольца равна: {RingS(a, b)}");

}

}

}

}



Лабораторная работа 12

Тема. Строки

Цель: изучить встроенную поддержку С# работы со строками, изучить множество встроенных методов для сравнения, поиска, сортировки и управления строковыми значениями

Задание 1. Для закрепления теоретического материала по теме Строки необходимо изучить теоретический материал и выполнить предложенные задания.

При решении задач использовать тип string*.*

Задание. Разработать программу, которая позволяет ввести строку с экрана и для введенной строки s:

1. подсчитывает общее число вхождений символов х и y;

static void Print(char[] arr)

{

foreach (char elem in arr)

{

Console.Write(elem);

}

Console.WriteLine();

}

static void Main(string[] args)

{

string s;

char x, y;

char[] arr;

int count = 0;

x = 'а';

y = 'п';

Console.WriteLine("Введите строку: ");

s = Convert.ToString(Console.ReadLine());

arr = s.ToCharArray();

Console.WriteLine("Ваш массив символов: ");

Print(arr);

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

{

if (arr[i] == x || arr[i] == y)

{

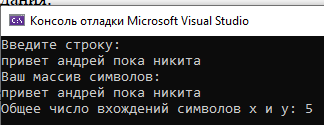
count++;

}

}

Console.WriteLine($"Общее число вхождений символов х и y: {count}");

}



1. определяет, какой из двух заданных символов встречается в строке чаще всего;

static void Print(char[] arr)

{

foreach (char elem in arr)

{

Console.Write(elem);

}

Console.WriteLine();

}

static void Main(string[] args)

{

string s;

char x, y;

char[] arr;

int countx = 0;

int county = 0;

x = 'а';

y = 'п';

Console.WriteLine("Введите строку: ");

s = Convert.ToString(Console.ReadLine());

arr = s.ToCharArray();

Console.WriteLine("Ваш массив символов: ");

Print(arr);

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

{

if (arr[i] == x)

{

countx++;

}

if (arr[i] == y)

{

county++;

}

}

if (countx > county)

{

Console.WriteLine($"Символ {x} встречается чаще, чем символ {y}");

}

if (countx < county)

{

Console.WriteLine($"Символ {y} встречается чаще, чем символ {x}");

}

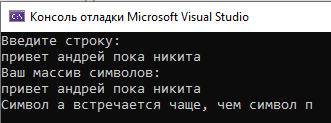
else

{

Console.WriteLine($"Символы встречаются одинаково.");

}

}



3) выводит на экран символы, которые наиболее часто встречается в строке;

4) выводит на экран символы, которые встречаются в строке только один раз;

5) определяет, имеются ли в строке два соседствующих одинаковых символа;

6) определяет, является ли строка палиндромом;

7) определяет, упорядочены ли по алфавиту символы строки;

static void Print(char[] arr)

{

foreach (char elem in arr)

{

Console.Write(elem);

}

Console.WriteLine();

}

static void Main(string[] args)

{

string s;

bool tf = true;

char[] arr\_sort;

char[] arr\_nonesort;

Console.WriteLine("Введите строку: ");

s = Convert.ToString(Console.ReadLine());

arr\_nonesort = s.ToCharArray();

arr\_sort = s.ToCharArray();

Array.Sort(arr\_sort);

Console.WriteLine("Отсортированный массив: ");

Print(arr\_sort);

for (int i = 0; i < arr\_sort.Length; i++)

{

if (arr\_sort[i] == arr\_nonesort[i])

{

tf = true;

}

else

{

tf = false;

}

}

if (tf == true)

{

Console.WriteLine("Символы строки упорядочены в алфавитном порядке");

}

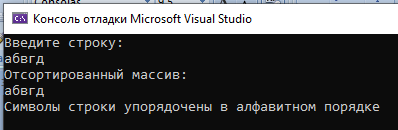
else

{

Console.WriteLine("Символы строки не упорядочены в алфавитном порядке");

}

}



8) подсчитывает количество букв в строке;

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите строку s:");

string s = Console.ReadLine( );

char[] a;

int sh1 = 0;

a = s.ToCharArray();

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

{

if (char.IsLetter(a[i]))

{

sh1++;

}

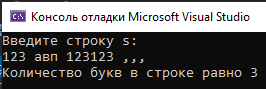
}

Console.WriteLine($"Количество букв в строке равно {sh1}");

}

}

}



9) подсчитывает количество цифр в строке;

static void Print(char[] arr)

{

foreach (char elem in arr)

{

Console.Write(elem);

}

Console.WriteLine();

}

static void Main(string[] args)

{

string s;

int count = 0;

char[] arr;

Console.WriteLine("Введите строку: ");

s = Convert.ToString(Console.ReadLine());

arr = s.ToCharArray();

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

{

if (char.IsDigit(arr[i]))

{

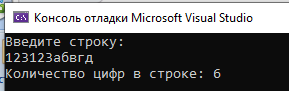
count++;

}

}

Console.WriteLine($"Количество цифр в строке: {count}");

}



10) подсчитывает сумму всех содержащихся в строке цифр;

static void Print(char[] arr)

{

foreach (char elem in arr)

{

Console.Write(elem);

}

Console.WriteLine();

}

static void Main(string[] args)

{

string s;

int num = 0;

char[] arr;

Console.WriteLine("Введите строку: ");

s = Convert.ToString(Console.ReadLine());

arr = s.ToCharArray();

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

{

if (char.IsDigit(arr[i]))

{

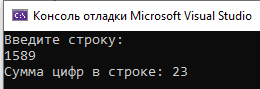
num += arr[i] - '0';

}

}

Console.WriteLine($"Сумма цифр в строке: {num}");

}



11) выводит на экран последовательность символов, расположенных до первого двоеточия;

static void Print(char[] arr)

{

foreach (char elem in arr)

{

Console.Write(elem);

}

Console.WriteLine();

}

static void Main(string[] args)

{

string s;

int num = 0;

char[] arr;

char dvoethochie = ':';

Console.WriteLine("Введите строку: ");

s = Convert.ToString(Console.ReadLine());

arr = s.ToCharArray();

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

{

if (dvoethochie == arr[i])

{

break;

}

else

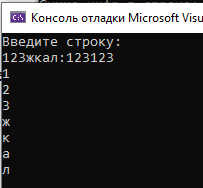
{

Console.WriteLine(arr[i]);

}

}

}



12) выводит на экран последовательность символов, расположенных после последнего двоеточия;

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите строку S: ");

string s = Console.ReadLine();

Console.Write("Строка после последнего двоеточия = ");

int index = s.LastIndexOf(':');

if (index >= 0)

{

Console.WriteLine(s.Substring(index+1));

}

else

{

Console.WriteLine(s);

}

}

}

}

13) выводит на экран последовательность символов, расположенных между круглыми скобками (считается, что в строке ровно одна пара круглых скобок);

using System;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

Console.Write("Введите строку S: ");

string s = Console.ReadLine();

Console.Write("Между скобками = ");

int index = s.IndexOf('(');

int index2 = s.LastIndexOf(')');

if (index >= 0 && index2 >= 0)

{

Console.WriteLine(s.Substring(index+1,index2 - (index + 1)));

}

else

{

Console.WriteLine(s);

}

}

}

}

Лабораторная работа

Тема. Строки

Цель: изучить встроенную поддержку С# работы со строками, изучить множество встроенных методов для сравнения, поиска, сортировки и управления строковыми значениями

Задание 2. Для закрепления теоретического материала по теме Строки необходимо изучить теоретический материал и выполнить предложенные задания.

При решении задач использовать класс StringBuilder

Задание. Разработать программу, которая позволяет ввести строку с экрана и выполняет следующие действия с введенной строкой:

1. вставляет в строку символ x после каждого вхождения символа y;

static void Main(string[] args)

{

Console.WriteLine("Введите любую строку: ");

StringBuilder user\_str = new StringBuilder(Console.ReadLine());

Console.WriteLine("Введите символ, который хотите вставлять после каждого символа вашей строки: ");

string x = Console.ReadLine();

Console.WriteLine("Введите символ, после которого надо вставить предыдущий символ: ");

char y = Convert.ToChar(Console.ReadLine());

for (int i = 0; i < user\_str.Length; i++)

{

if (user\_str[i] == y)

{

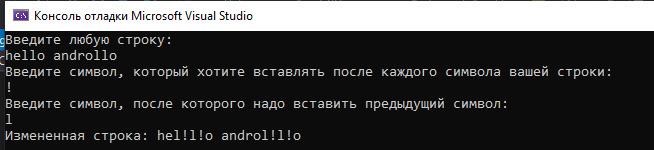
user\_str.Insert(i + 1, x);

}

}

Console.WriteLine($"Измененная строка: {user\_str}");

}



1. вставляет в строку подстроку x после каждого вхождения подстроки y;

static void Main(string[] args)

{

Console.WriteLine("Введите любую строку: ");

StringBuilder user\_str = new StringBuilder(Console.ReadLine());

Console.WriteLine("Введите подстроку: ");

string x = Console.ReadLine();

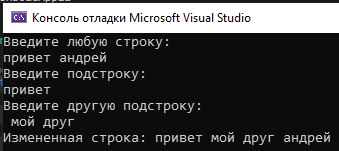
Console.WriteLine("Введите другую подстроку: ");

string y = Console.ReadLine();

user\_str.Replace(x, x + y);

Console.WriteLine($"Измененная строка: {user\_str}");

}



1. удваивает каждое вхождение заданного символа x;

static void Main(string[] args)

{

Console.WriteLine("Введите любую строку: ");

StringBuilder user\_str = new StringBuilder(Console.ReadLine());

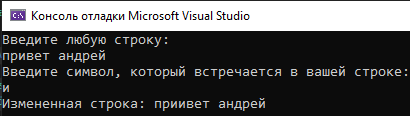
Console.WriteLine("Введите символ, который встречается в вашей строке: ");

string x = Console.ReadLine();

user\_str.Replace(x, x + x);

Console.WriteLine($"Измененная строка: {user\_str}");

}



1. удваивает каждое вхождение заданной подстроки x;

static void Main(string[] args)

{

Console.WriteLine("Введите любую строку: ");

StringBuilder user\_str = new StringBuilder(Console.ReadLine());

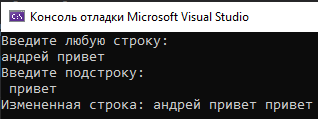
Console.WriteLine("Введите подстроку: ");

string x = Console.ReadLine();

user\_str.Replace(x, x + x);

Console.WriteLine($"Измененная строка: {user\_str}");

}



1. удаляет среднюю букву, если длина строки нечетная, и две средних, если длина строки четная;

static void Main(string[] args)

{

Console.WriteLine("Введите любую строку: ");

StringBuilder user\_str = new StringBuilder(Console.ReadLine());

StringBuilder edit\_str = new StringBuilder();

string tmp\_user\_str = Convert.ToString(user\_str);

string[] txtMass;

txtMass = tmp\_user\_str.Split(' ');

for (int i = 0; i <= txtMass.Length-1; i++)

{

char[] txtMassChar = txtMass[i].ToCharArray();

Console.WriteLine(txtMassChar);

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

{

int tmp\_bool = txtMassChar.Length % 2;

int middle = txtMassChar.Length / 2;

if (tmp\_bool == 0)

{

txtMass[i].ToString();

txtMass[i].Remove(txtMassChar[middle]);

}

else

{

txtMass[i].ToString();

txtMass[i].Remove(txtMassChar[middle]);

}

}

edit\_str.Append(txtMass[i]);

}

Console.WriteLine($"Измененная строка: {edit\_str}");

}