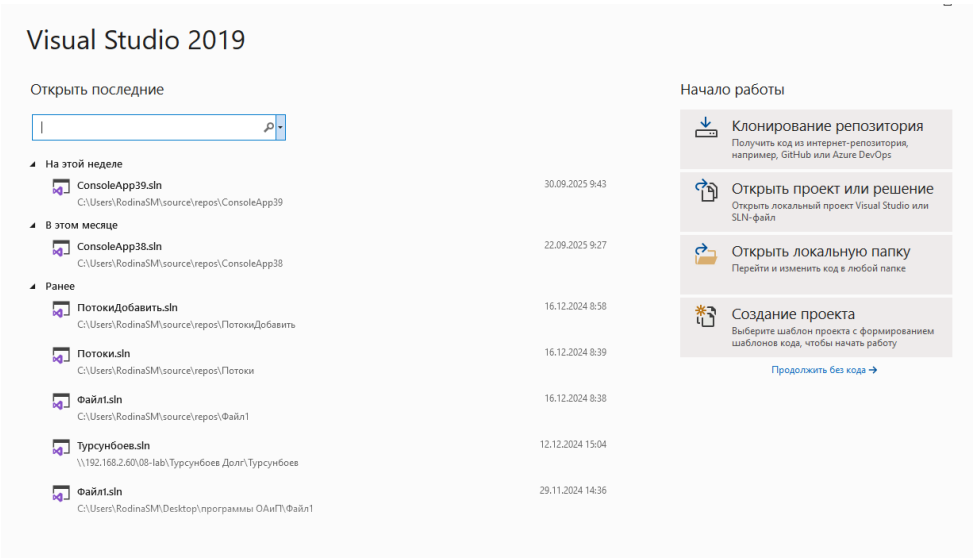
Лабораторная работа № 1

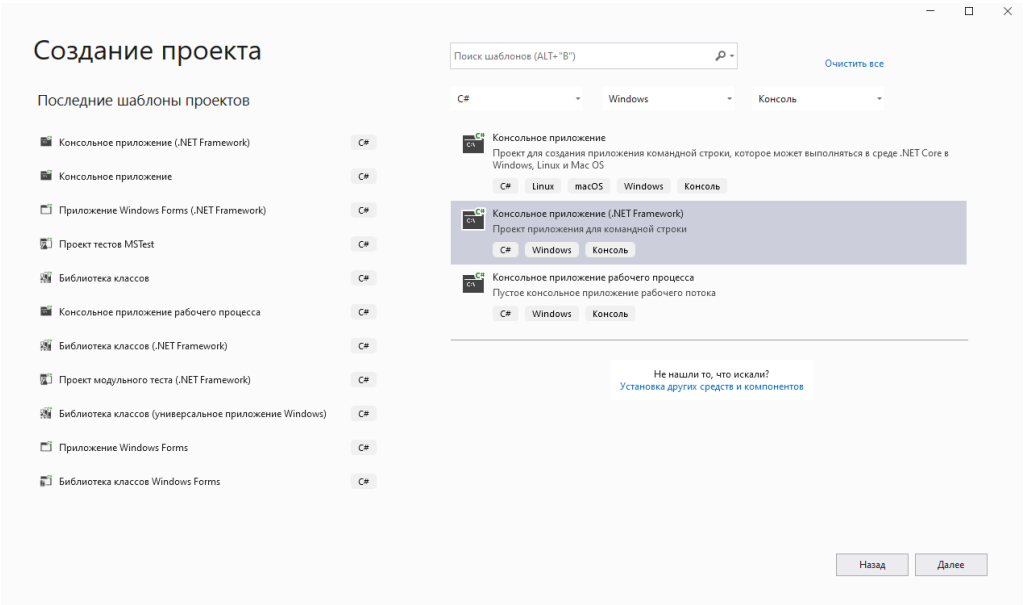
Цель работы: Научиться создавать консольные приложения. Работать с арифметическими выражениями.

Задание 1.

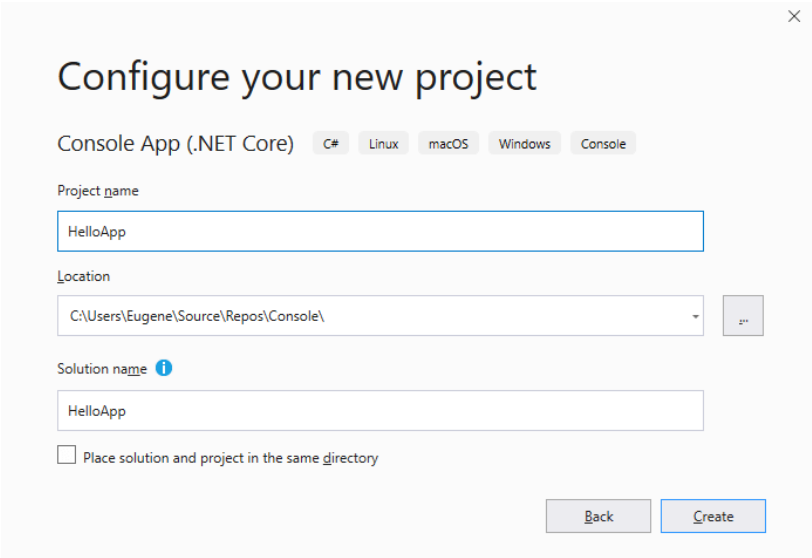
Создадим первую программу. Она будет простенькой. Вначале откроем Visual Studio. На стартовом экране выберем **Create a new project** (Создать новый проект)



На следующем окне в качестве типа проекта выберем Консольное приложение (.NET Framwork), то есть мы будем создавать консольное приложение на языке C#

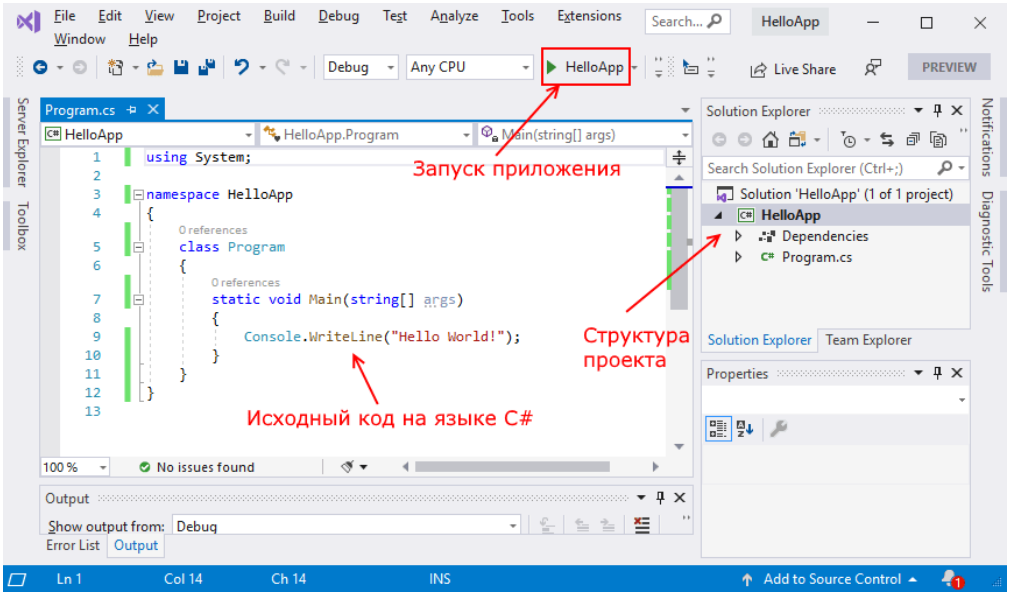


Далее в поле Project Name дадим проекту какое-либо название. Пусть это будет HelloApp. А также укажем папку, где будет располагаться проект. И нажмем Create (Создать).



Здесь в центре мы выберем пункт Console App (.NET Framework),

После этого Visual Studio создаст и откроет нам проект:



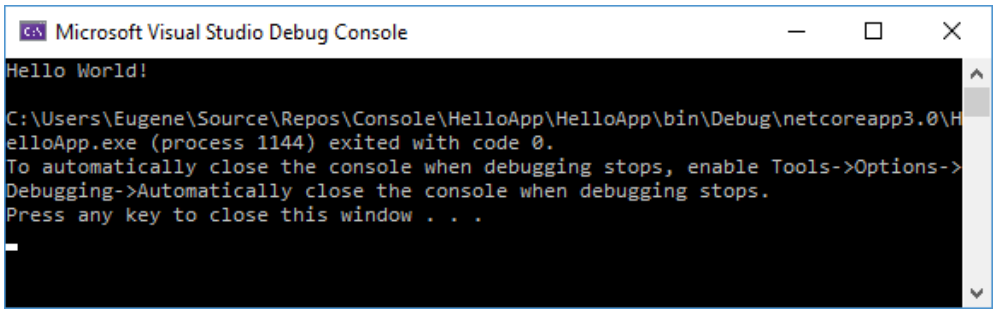
В большом поле в центре, которое по сути представляет текстовый редактор, находится сгенерированный по умолчанию код C#. Впоследствии мы изменим его на свой.

Справа находится окно Solution Explorer, в котором можно увидеть структуру нашего проекта.

Далее идет непосредственно сам файл кода программы Program.cs. Как раз этот файл и открыт в центральном окне. Вначале разберем, что весь этот код представляет:

```
using System; // подключаемое пространство имен

namespace HelloApp // объявление нового пространства имен
{
    class Program   // объявление нового класса
    {
        static void Main(string[] args) // объявление нового метода
        {
            Console.WriteLine("Hello World!"); // действия метода

        } // конец объявления нового метода

    } // конец объявления нового класса

} // конец объявления нового пространства имен
```

Теперь мы можем запустить на выполнение с помощью клавиши F5 или с панели инструментов, нажав на зеленую стрелку. И если вы все сделали правильно, то при запуске приложения мы сможем ввести свое имя, и затем оно будет выведено на консоль.



Теперь изменим весь этот код на следующий:

```
using System;

namespace HelloApp
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Введите свое имя: ");
            string name = Console.ReadLine();        // вводим имя
            Console.WriteLine($"Привет {name}");     // выводим имя на
консоль
            Console.ReadKey();
        }
    }
}
```

По сравнению с автоматически сгенерированным кодом внесем несколько изменений. Теперь в методе Main первой строкой выводится приглашение к вводу.

```
1        Console.Write("Введите свое имя: ");
```

На второй строке определяется строковая переменная name, в которую пользователь вводит информацию с консоли:

```
1        string name = Console.ReadLine();
```

То есть с помощью метода Console.ReadLine() мы можем считать с консоли строку.

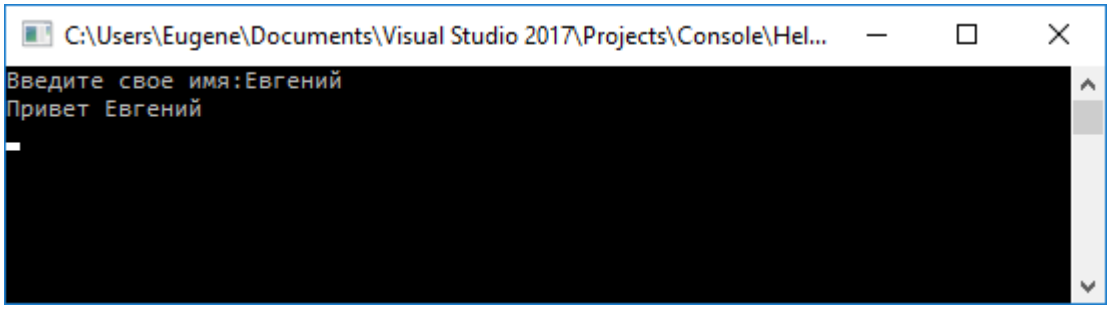
Затем введенное имя выводится на консоль:

```
1        Console.WriteLine($"Привет {name}");
```

Чтобы ввести значение переменной name внутрь выводимой на консоль строки, применяются фигурные скобки {}. То есть при выводе строки на консоль выражение {name} будет заменяться на значение переменной name - введенное имя.

Однако, чтобы можно было вводить таким образом значения переменных внутрь строки, перед строкой указывается знак доллара $.

Теперь протестируем проект, запустив его на выполнение, также нажав на F5 или зеленую стрелочку.



Итак, мы создали первое приложение. Вы его можете найти на жестком диске в папке проекта в каталоге bin/Debug. Оно будет называться по имени проекта и иметь расширение exe (для более ранних версий .NET Core это файл с расширением dll). И затем этот файл можно будет запускать без Visual Studio, а также переносить его на другие компьютеры, где есть .NET Core.

Задания для самостоятельного выполнения!

Наберите и выполните код программы

```
static void Main(string[] args)

{

   string stringToShow1, stringToShow2;



   string surname = "Ваша фамилия";

   string name = "Ваше имя";

   string otchestvo = "Ваше отчество";



   int age = Ваш возраст;

   double weight = ваш вес;



   stringToShow1 = surname + " " + name + " " + otchestvo + ", возраст " + age + ", вес
" + weight;



   surname = "Фамилия вашего друга";

   name = "Имя вашего друга";

   otchestvo = "Отчество вашего друга";



   age = ???;

   weight = ???;



   stringToShow2 = surname + " " + name + " " + otchestvo + ", возраст " + age +  ", вес
" + weight;



   System.Console.WriteLine(stringToShow1);

   System.Console.WriteLine(stringToShow2);
```

Что получилось???

Измените вывод таким образом, чтобы вывести отдельно на каждую строчку каждый параметр по каждому человеку отдельно.

Задание 2.

Приложения c# для расчетов по формулам, консольный ввод-вывод.

Константы

Это неизменяемые в процессе выполнения программы величины.

Целые константы - наиболее распространенный тип int. Это целое число, которое может быть отрицательным, положительным или нулем -12, 5, 0 (все целые со знаком 32 бита). Их можно записывать с суффиксом -12L (длинное целое 64 бита), 5u (целое без знака 8 бит)

Вещественные константы с фиксированной точкой. При записи константы типа float(32 бита) необходимо, чтобы за значением шел суффикс символ f или F 1.2, -1.234, при записи константы типа double (64 бита) можно записать суффикс «d» или «D», но это не является обязательным условием: 1234.5678, 12.3d. Дробная часть отделяется от целой части точкой.

Вещественные константы с плавающей точкой. При записи константы типа float(32 бита) необходимо, чтобы за значением шел суффикс символ f или F: 1.2E-3f (число 0.0012), при записи константы типа double (64 бита) -1.34E5 (число -134000) наличие суффикса не требуется.

Символьные константы. Символьная константа char может представлять собой 16-битный символ Unicode ('a') или управляющие символы (возврат каретки ('\r'), перевод страницы ('\f'), горизонтальную табуляцию ('\t'), и другие), заключенный в апострофы.

Строковые константы - это последовательность символов, заключенная в кавычки, или константы string. Строка, состоящая из символов, например "Ура!\n Сегодня \"Основы алгоритмизации\"!!!"

Логическая константа. Задается одним из двух значений true («истина») или false («ложь»). Используется в C# в логических выражениях, операторах условного перехода.

Именованные константы. Применяются для того, чтобы вместо значений констант, использовать в программе их имена, например константа p вещественная одинарной точности

const float p = 3.14159f

Переменные

Переменная - именованная область памяти, для хранения данных определенного типа. При выполнении программы значение переменной величины можно изменять. Все

переменные должны быть описаны явно, при описании переменной задается ее значение и тип. При объявлении переменной может быть задано начальное значение.

Имя переменной может содержать буквы, цифры и символ подчеркивания. Прописные и строчные буквы различаются. Например, переменные Long, LONG, long - три разных переменные.

Имя переменной может начинаться с буквы или знака подчеркивания, но не цифры. Имя переменной не должно совпадать с ключевыми словами. Не рекомендуется начинать имя с двух подчеркиваний (такие имена зарезервированы для служебного использования).

Правильные имена переменных: MaxLen, iMaxLen, Max_Len

Неправильные имена переменных: 2Len, Le#

Примеры описания переменных:

int a = -14; // числовая целая 32 бита

float c = -0.00151f; // числовая вещественная 32

// бита

double i = 1234.56789; // числовая вещественная 64

// бита

bool l = false; // логическая 16 бит

string name = "Petrov"; // строковая

Выражение - состоит из одного или более операндов (которые могут быть переменными, константами, функциями или символьными значениями), знаков операций и круглых скобок.

Примеры выражений:

2 * 2 + 1 полученное значение 5

1 / 2 - 3 полученное значение -3

1 / 2 - 3 полученное значение -2.5

Присвоение значения переменной представляет оператор присваивания (знаки основных операций приведены в таблице 1.2) : y = 2*x*x + 3*x - 1.

В этом примере сначала производятся вычисления правой части оператора присваивания « = », а затем полученное значение присваивается переменной y. Для текстовых данных выражение можно записать в следующем виде:

string Tex = "ГБПОУ" + "ОЗЖТ";

В этом примере строки по правую сторону от оператора присваивания объединяются, чтобы получить строку " ГБПОУОЗЖТ ", которая затем присваивается переменной Tex.

Таблица 1.2 Знаки операций

Знак операции	Название
+	Сложение
-	Вычитание
*	Умножение
/	Деление
%	Остаток от деления

Если в арифметических выражениях используются целые числа, то результатом вычислений будет целое число, и любой остаток от деления будет отброшен. Для получения остатка можно использовать соответствующую операцию %, например 10 % 3 возвращает остаток от целочисленного деления, равный 1.

Когда в арифметических выражениях используются числа с плавающей точкой, то результатом деления 10f / 3f будет число 3,333333.

Математические функции.

C# содержит большое количество встроенных математических функций, которые реализованы в классе Math пространства имен System.

Рассмотрим краткое описание некоторых математических функций, подробнее с ними можно познакомиться в справочной системе VS или технической документации. Особое внимание следует обратить на типы операндов и результатов, т. к. каждая функция может иметь несколько перегруженных версий.

Замечание. Использование нескольких функций с одним и тем же именем, но с различными типами параметров, называется перегрузкой функции. Например, функция Math.Abs(), вычисляющая модуль числа, имеет 7 перегруженных

версий: double Math.Abs (double x), float Math.Abs (float x), int Math.Abs(int x), и т. д. (таблица 1.3)

Таблица 1.3 Математические функции

№	Название	Описание
1.	**Math.Abs***(аргумент)*	Модуль
2.	**Math.Ceiling*(аргумент)***	Округление до большего целого
3.	**Math.Cos** *(* **аргумент** *)*	Косинус
4.	**Math.E**	Число e
5.	**Math.Exp***(аргумент)*	Экспонента

6.	**Math.Floor***(аргумент)*	Округление до меньшего целого
7.	**Math.Log***(аргумент)*	Натуральный логарифм
8.	**Math.Log10***(аргумент)*	Десятичный логарифм
9.	**Math.Max***(аргумент1, аргумент2)*	Максимум из двух значений
10.	**Math.Min** *(аргумент1, аргумент2)*	Минимум из двух значений
11.	**Math.PI**	Число
12.	**Math.Pow***(аргумент1, аргумент2)*	Возведение в степень
13.	**Math.Roundl***(аргумент)*	Простое округление
	Math.Round*(аргумент,число)*	Округление до заданного числа цифр
14.	**Math.Sign***(аргумент)*	Знак числа
15.	**Math.Sin***(аргумент)*	Синус
16.	**Math.Sqrt***(аргумент)*	Квадратный корень
17.	**Math.Tan** *(аргумент)*	Тангенс

Пример 2. Вычислить значения функции при x = 2,5

```
using System;
using System.Collections.Generic;
using System.Linq; using System.Text;
namespace ConsoleApplication1
{
class Example2 // начало описания класса
// Example2
{
static void Main()
{
double p = 3.14159; double x = 2.5;
double y = Math.Cos(p * x)/(1 + x*x);
Console.WriteLine();
Console.WriteLine(" x = {0} \t y = {1} ",x, y);
}
}
}
```

Эта программа выводит следующее окно с результатом:

Замечание. Функция Console.WriteLine() выводит на экран пустую строку. Это сделано для более комфортной работы

Организация ввода-вывода данных.

Программа при вводе данных и выводе результатов взаимодействует с внешними устройствами. Совокупность стандартных устройств ввода (клавиатура) и вывода (экран) называется консолью. В языке C# нет операторов ввода и вывода. Вместо них для обмена данными с внешними устройствами используются специальные объекты. В частности, для работы с консолью используется стандартный класс Console, определенный в пространстве имен System.

Ввод данных

Для ввода данных обычно используется метод ReadLine, реализованный в классе Console. Особенностью данного метода является то, что в качестве результата он возвращает строку (string).

Пример:

```
static void Main()
{
string s = Console.ReadLine(); Console.WriteLine(s);
}
```

Для того чтобы получить числовое значение необходимо воспользоваться преобразованием данных.

Пример:

```
static void Main()
{
string s = Console.ReadLine();
int x = int.Parse(s); // преобразование строки в число
Console.WriteLine(x);
}
```

Или сокращенный вариант:

```
static void Main()
{
//преобразование введенной строки в число
int x = int.Parse(Console.ReadLine());
Console.WriteLine(x);
}
```

Для преобразования строкового представления целого числа в тип int мы используем метод int.Parse(), который реализован для всех числовых типов данных. Таким образом, если нам потребуется преобразовать строковое представление в вещественное, мы можем воспользоваться методом float.Parse() или double. Parse(). В случае, если соответствующее преобразование выполнить невозможно, то выполнение программы прерывается и генерируется исключение System.FormatExeption (входная строка имела неверный формат).

Вывод данных

В приведенных выше примерах мы уже рассматривали метод WriteLine, реализованный в классе Console, который позволяет организовывать вывод данных на экран. Однако существует несколько способов применения данного метода:

Console.WriteLine(x);

на экран выводится значение идентификатора x

Console.WriteLine("x=" + x +"y=" + y);

на экран выводится строка, образованная последовательным слиянием строки "x=", значения x, строки "y=" и значения y

Console.WriteLine ("x={0} y={1}", x, y);

на экран выводится строка, формат которой задан первым аргументом метода, при этом вместо параметра {0} выводится значение x, а вместо {1} - значение

Если использовать при выводе вместо метода WriteLine метод Write, вывод будет выполняться без перевода строки.

Использование управляющих последовательностей.

Управляющие символы (таблица 1.4).

Код	Значение
\b	Возврат на одну позицию
\f	Подача страницы (для перехода к началу следующей страницы)
\n	Новая строка
\r	Возврат каретки
\t	Горизонтальная табуляция
\"	Двойная кавычка
\'	Одинарная кавычка (апостроф)
\\	Обратная косая черта
\v	Вертикальная табуляция
\a	Звуковой сигнал (звонок)
\?	Вопросительный знак
N	Восьмеричная константа (где *N* — это сама восьмеричная константа)
\x*N*	Шестнадцатеричная константа (где *N* — это сама шестнадцатеричная константа)

Управляющей последовательностью называют определенный символ, предваряемый обратной косой чертой. Данная совокупность символов интерпретируется как одиночный символ и используется для представления кодов символов, не имеющих графического обозначения (например, символа перевода курсора на новую строку) или символов, имеющих специальное обозначение в символьных и строковых константах (например, апостроф). Рассмотрим управляющие символы:

Пример 3. Вывести сообщение о версии установленной операционной системы, текущую дату и время.

```
using System;
using System.Collections.Generic;
using System.Linq; using System.Text;
namespace ConsoleApplicationl
{
class Program
{
static void Main(string[] args)
{
// вывести версию операционной системы
OperatingSystem os = System.Environment.OSVersion;
Console.WriteLine("Platform: {0}",os.Platform);
System.Console.WriteLine("The current date and time is
" + System.DateTime.Now);
// дата и время System.Console.ReadLine();
}
}
}
```

Пример 4. Использование консольного ввода для вычисления значений функции

```
using System;
using System.Collections.Generic;
using System.Linq; using System.Text;
namespace lab0
{
class Program
{
static void Main(string[] args)
{
System.Console.WriteLine("The current date and time is
" + System.DateTime.Now);
double pi = 3.14159;
Console.WriteLine("Input x =\r");
double x = Convert.ToDouble(Console.ReadLine());
double y = Math.Cos(pi * x)/(1 + x*x);
Console.WriteLine(" x = {0} \t y = {1} ",x,y);
Console.ReadKey();
}
}
}
```

3. Практическая часть.

Задание 1. Составить процедуру для выполнения расчетов функции, значения задавать в диалоге с использованием метода Console.ReadLine() (пример 4) см. таблицу 1.5 и таблицу 1.6;

Таблица 1.5

№ вар.	Задание	№ вар.	Задание
1	$R = 3t^2 + 3t^5 + 4.9$	16	$S = \sqrt{\cos 4y^2 + 7{,}151}$
2	$K = \ln\left(p^2 + y^3\right) + e^p$	17	$N = 3y^2 + \sqrt{y+1}$
3	$G = n(y + 3{,}5) + \sqrt{y}$	18	$Z = 3y^2 + \sqrt{y^3 + 1}$
4	$D = 9{,}8a^2 + 5{,}52\cos t^5$	19	$P = n\sqrt{y^3 + 1{,}09g}$
5	$L = 1{,}51\cos x^2 + 2x^3$	20	$U = e^{k+y} + \mathrm{tg}x\sqrt{y}$
6	$M = \cos 2y + 3{,}6e^x$	21	$P = e^{y+5{,}5} + 9{,}1h^3$
7	$N = m^2 + 2{,}8\lvert m\rvert + 0{,}55$	22	$T = \sin(2u)\ln\left(2y^2 + \sqrt{x}\right)$
8	$T = \sqrt{\lvert 6y^2 - 0{,}1y + 4\rvert}$	23	$G = e^{2y} + \sin(f)$
9	$V = \ln(y + 0{,}95) + \sin x^4$	24	$F = 2\sin\left(0{,}214y^5 + 1\right)$
10	$U = e^y + 7{,}355k^2 + \sin^2 x$	25	$G = e^{2y} + \sin\left(f^2\right)$
11	$S = 9{,}756y^7 + 2\mathrm{tg}x$	26	$Z = \sin\left(p^2 + 0{,}4\right)^3$
12	$K = 7t^2 + 3\sin x^3 + 9{,}2$	27	$W = 1{,}03v + e^{2y} + \mathrm{tg}\lvert x\rvert$
13	$E = \sqrt{\lvert 3y^2 + 0{,}5y + 4\rvert}$	28	$T = e^{y+h} + \sqrt{\lvert 6{,}4y\rvert}$
14	$R = \left\lvert\sqrt{\sin^2 y + 6{,}835} + e^x\right\rvert$	29	$N = 3y^2 + \sqrt{\lvert y+1\rvert}$
15	$H = \sin y^2 - 2{,}8y + \sqrt{\lvert y\rvert}$	30	$W = e^{y+r} + 7{,}2\sin r$

Таблица 1.6

№ вар.	Выражение	№ вар.	Выражение
1	$G = \frac{e^{2y} + \sin f}{\ln(3{,}8y + f)}$	16	$W = \frac{4t^3 + \ln r}{e^{y+r} + 7{,}2\sin r}$
2	$F = \ln d + \frac{3{,}5d^2 + 1}{\cos 2y}$	17	$H = \frac{y^2 - 0{,}8y + \sqrt{y}}{23{,}1n^2 + \cos n}$
3	$U = \frac{\ln(k - y) + y^4}{e^y + 2{,}355k^2}$	18	$R = \frac{\sqrt{\sin^2 y + 6{,}835}}{\ln(y + k) + 3y^2}$
4	$G = \frac{9{,}33w^3 + \sqrt{w}}{\ln(y + 3{,}5) + \sqrt{y}}$	19	$E = \frac{\ln(0{,}7y + 2q)}{\sqrt{3y^2 + 0{,}5y + 4}}$
5	$D = \frac{7{,}8a^2 + 3{,}52t}{\ln(a + 2y) + e^y}$	20	$K = \frac{2t^2 + 3l + 7{,}2}{\ln y + e^{2l}}$
6	$L = \frac{0{,}81\cos i}{\ln y + 2i^3}$	21	$Q = \frac{\sqrt{k + 2{,}6p\sin k}}{x - d^3}$
7	$N = \frac{m^2 + 2{,}8m + 0{,}355}{\cos 2y + 3{,}6}$	22	$S = \frac{4{,}351y^3 + 2t\ln t}{\sqrt{\cos 2y + 4{,}351}}$
8	$T = \frac{2{,}37\sin(t + 1)}{\sqrt{4y^2 - 0{,}1y + 5}}$	23	$R = \frac{\sin^2 y + 0{,}3d}{e^y + \ln d}$
9	$V = \frac{(y + 2w)^3}{\ln(y + 0{,}75)}$	24	$U = \frac{\ln(2k + 4{,}3)}{e^{k+y} + \sqrt{y}}$
10	$Z = \frac{2t + y\cos t}{\sqrt{y + 4{,}831}}$	25	$L = \cos^2 c + \frac{3t^2 + 4}{\sqrt{c + t}}$
11	$D = y^2 + \frac{0{,}5n + 4{,}8}{\sin y}$	26	$T = \frac{\sin 2u}{\ln(2y + u)}$
12	$R = \frac{\sin(2t + 1)^2 + 0{,}3}{\ln(t + y)}$	27	$Z = \frac{\sin(p + 0{,}4)^2}{y^2 + 7{,}325p}$
13	$A = \frac{\sin(2y + h) + h^2}{e^k + y}$	28	$W = \frac{0{,}004v + e^{2y}}{e^{\frac{y}{2}}}$
14	$P = \frac{e^{y+2.5} + 7{,}1h^3}{\ln\sqrt{y + 0{,}04h}}$	29	$T = \frac{0{,}355h^2 - 4{,}355}{e^{y+h} + \sqrt{2{,}7y}}$
15	$F = \frac{2\sin(0{,}354y + 1)}{\ln(y + 2j)}$	30	$N = \frac{3y^2 + \sqrt{y + 1}}{\ln(p + y) + e^p}$