Главное управление образования Гомельского облисполкома

Учреждение образования

«Гомельский государственный машиностроительный колледж»

Учебная дисциплина

«Конструирование программ и языки программирования»

**Инструкция**

по выполнению лабораторной работы №3

«Разработка программ с использованием операторов выбора, цикла и

передачи управления»

Гомель 2018

Составитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Гаврилова В.М.

Обсуждено и одобрено на заседании цикловой комиссии «Программное обеспечение информационных технологий»

Протокол от «01» сентября 2018 № 1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.Ю. Лавская

**Лабораторная работа №3**

**Тема работы: «Разработка программ с использованием операторов выбора, цикла и передачи управления»**

1. **Цель работы**

Сформировать умения разрабатывать программы с использованием операторов выбора, цикла, передачи управления.

1. **Задание**

**Задание 1.**

**1**. Ввести три числа A, B, C. Найти сумму двух наибольших из них. Если пользователь введёт числа так, что среди них нельзя будет определить два наибольших - выдать соответствующее сообщение.

**2**. Ввести три числа A, B, C. Вывести вначале наименьшее, а затем наибольшее из данных чисел. Если пользователь введёт числа так, что среди них нельзя будет определить одно наименьшее/наибольшее - выдать соответствующее сообщение.

**3**. Ввести три числа A, B, C. Вывести среднее из них (то есть число, расположенное между наименьшим и наибольшим). Если пользователь введёт числа так, что среди них нельзя будет определить среднее - выдать соответствующее сообщение.

**4**. Ввести три целых числа A, B, C. Найти количество положительных и количество отрицательных чисел в исходном наборе.

**5**. Даны три числа: A, B, и C. Удвоить эти числа, если , и заменить их абсолютными значениями, если это не так.

**6**. Заданы натуральные числа: A, B, C и D. Выяснить, является ли хотя бы одно из чисел C или D остатком от деления A на B.

**7**. Даны натуральные числа A, B, C и D. Выяснить, можно ли прямоугольник со сторонами A и B уместить внутри прямоугольника со сторонами C и D так, чтобы каждая сторона одного прямоугольника была параллельна или перпендикулярна стороне другого прямоугольника.

**8**. Решить неравенство для значений А и В.

**9**. Даны 2 числа. Вывести первое число, если оно больше второго, и оба числа, если это не так.

**10**. Определить квадрат наибольшего из четырёх чисел А, В,С, D, значения которых заданы.

**11**. Даны целые числа a, b, c, являющиеся сторонами некоторого треугольника. Проверить истинность высказывания: «Треугольник со сторонами a, b, c является прямоугольным».

**12**. Дано трехзначное число. Проверить истинность высказывания: «Цифры данного числа образуют возрастающую или убывающую последовательность».

**13**. Дано целое положительное число. Проверить истинность высказывания: «Данное число является нечетным трехзначным».

**14**. Даны координаты поля шахматной доски x, y (целые числа, лежащие в диапазоне 1–8). Учитывая, что левое нижнее поле доски (1, 1) является черным, проверить истинность высказывания: «Данное поле является белым».

**15**. Ввести номер года (положительное целое число). Определить количество дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются).

**16**. Ввести целочисленные координаты трех вершин прямоугольника, стороны которого параллельны координатным осям. Найти координаты его четвертой вершины. Если пользователь введёт координаты точек так, что нельзя получить прямоугольник со сторонами, параллельными координатным осям, вывести соответствующее сообщение.

**17**. На числовой оси расположены три точки: A, B, C. Определить, какая из двух последних точек (B или C) расположена ближе к A, и вывести эту точку и ее расстояние от точки A. Если пользователь введёт координаты B и C так, что они будут равноотстоящими от А, совпадать с А или между собой - выдать соответствующее сообщение.

**18**. Ввести четыре целых числа A, B, C, D, одно из которых отлично от трех других, равных между собой. Определить порядковый номер числа, отличного от остальных. Если пользователь введёт числа так, что они не будут соответствовать условию задачи - выдать сообщение об ошибке.

**19**. Ввести три числа А,В,С. Если среди них имеется хотя бы одно четное вывести максимальное из них, иначе - минимальное. Если пользователь введёт числа так, что среди них нельзя будет определить лишь одно максимальное/минимальное - выдать соответствующее сообщение.

**20**. Ввести три переменные вещественного типа: A, B, C. Если их значения упорядочены по возрастанию или убыванию, то удвоить их; в противном случае заменить знак каждой переменной на противоположный. Вывести новые значения переменных A, B, C.

## Задание 2

**1**. Ввести два целых числа N1 и N2. Если N1 >N2, найти сумму целых чисел в диапазоне N1 ... N2. Если N2 больше N1, найти сумму целых чисел в диапазоне N2... N1. Если N1 равно N2, вывести на экран соответствующее сообщение.

**2**. Осуществить ввод последовательности целых чисел, определить третье положительное число и подсчитать количество цифр в нем. Последовательность потенциально не ограничена, окончанием последовательности служит третье положительное число.

**3**. Осуществить ввод последовательности целых чисел, определить максимальное четное число, его порядковый номер и подсчитать сумму его цифр. Последовательность потенциально не ограничена, окончанием последовательности служит число 0. Если окажется, что чётных чисел в последовательности не было, вывести соответствующее сообщение.

**4**. Осуществить ввод последовательности целых чисел и сравнить, что больше, сумма положительных или произведение отрицательных. Последовательность потенциально не ограничена, окончанием последовательности служит число 0.

**5**. Осуществить ввод последовательности целых чисел и определить предпоследнее отрицательное число. Последовательность потенциально не ограничена, окончанием последовательности служит число 0. Если окажется, что в последовательности было менее двух отрицательных чисел, вывести соответствующее сообщение.

**6**. Осуществить ввод целого числа М. На промежутке от 1 до M найти все числа Армстронга. Натуральное число из n цифр называется числом Армстронга, если сумма его цифр, возведенных в n-ю степень, равна самому числу. Примеры: 153=13 + 53 + 33; 1634=14+64+34+44.

**7.** Ввести действительное число х и натуральное число n. Вычислить x\*(x - n)\*(x - 2\*n)\* \*(x - 3\*n)\*…\*(x - n2).

**8**. Осуществить ввод последовательности целых чисел. Определить, сколько из них и какие принимают наибольшее значение. Последовательность потенциально не ограничена, окончанием последовательности служит число 0.

**9**. Осуществить ввод последовательности целых чисел в количестве не меньшем двух. Вычислить сумму тех из них, порядковые номера которых - простые числа. Последовательность потенциально не ограничена, окончанием последовательности служит число 0.

**10**. Осуществить ввод последовательности целых чисел в количестве не меньшем трёх. Определить, сколько из них больше своих "соседей", т.е. предыдущего и последующего чисел. Последовательность потенциально не ограничена, окончанием последовательности служит число 0.

**11**. Ввести целое число N > 1 и две вещественные точки на числовой оси: A, B (A < B). Отрезок [A, B] разбит на N равных отрезков. Вывести H — длину каждого отрезка, а также значения функции f(x) = 1 – sin(x) в точках, разбивающих отрезок [A, B]: f(A), f(A + H), f(A + 2H), ..., f(B).

**12**. Ввести целое число N > 0 и вещественное a > 0. Последовательность вещественных чисел определяется следующим образом xn+1=(xn+a/xn)/2. Считая x0=a вывести первые N членов последовательности. Такой способ применяли еще в древнем Вавилоне для вычисления квадратного корня числа a. После выдачи последовательности распечатать значение квадратного корня из a, вычисленное стандартной функцией.

**13**. Ввести целое число N > 1. Последовательность чисел Фибоначчи FK (целого типа) определяется следующим образом: F1 =1, F2= 1, FK=FK-2 + FK-1, K = 3, 4, ... N. Вывести элементы F1, F2, …, FN.

**14**. Ввести целое число N > 0. Последовательность вещественных чисел AK определяется следующим образом: A0=1/0!, AK=1/K!, K=1, 2, ...,N. Вывести сумму последовательности. Примечание К! — К–факториал — обозначает произведение всех целых чисел от 1 до К.

**15**. Логистическое отображение (также известное, как квадратичное отображение или отображение Фейгенбаума) даётся формулой xn+1=r\*xn\*(1-xn). Считая x0=0.333 распечатать N первых элементов отображения. Величину r, принадлежащую интервалу (0..4) вводит пользователь. (При r>3.6 должна наблюдаться хаотическая последовательность). В качестве тестового примера построить последовательности при разных значениях r.

**16**. Ввести целое число N > 2. Последовательность целых чисел AK определяется следующим образом: A1=1, A2=2, A3=3, AK=AK–1 + AK–2 – 2\*AK–3, K = 4, 5, ... N. Вывести элементы A1, A2, ..., AN.

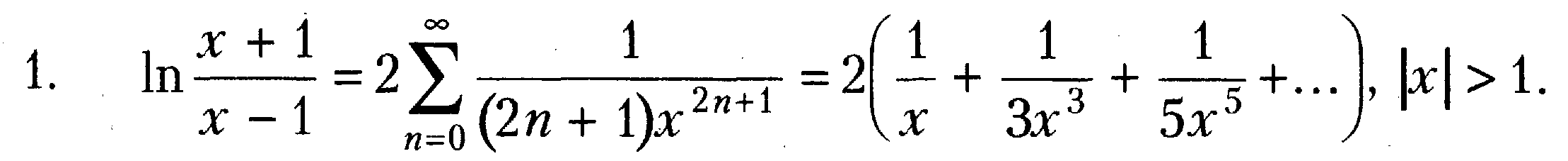
**17**. Ввести вещественное число X и целое число N > 0. Найти значение выражения X – X3/(3!) + X5/(5!) – ... + (–1)N\*X2\*N+1/((2\*N+1)!) , которое является приближенным значением функции sin в точке X. Отобразить сумму ряда и рассчитанное с помощью функции sin значения.

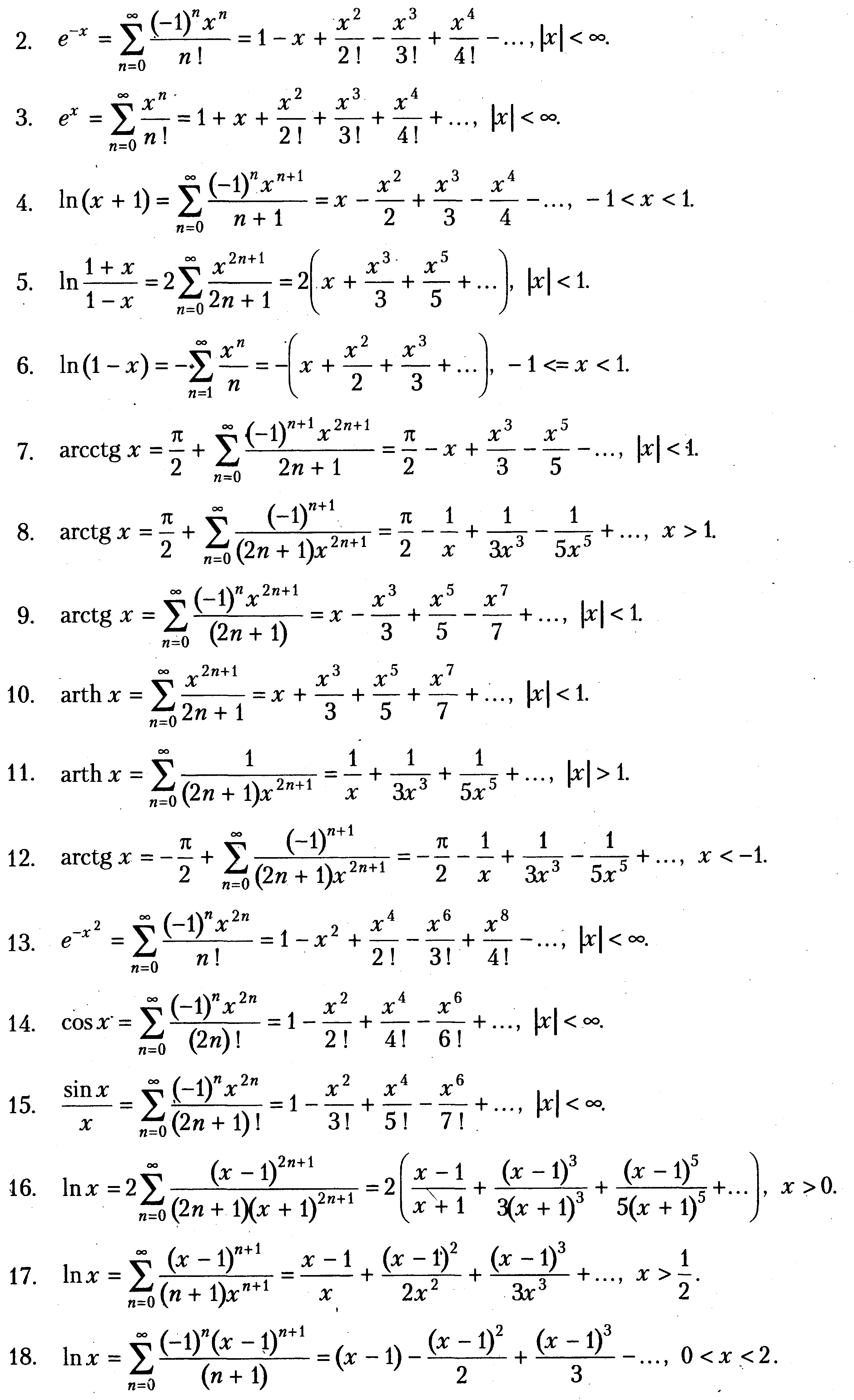
**18**. Ввести целое число N > 0. Найти квадрат данного числа, используя для его вычисления следующую формулу: N2=1 + 3 + 5 + ... + (2\*N – 1). После добавления к сумме каждого слагаемого выводить текущее значение суммы (в результате будут выведены квадраты всех целых чисел от 1 до N).

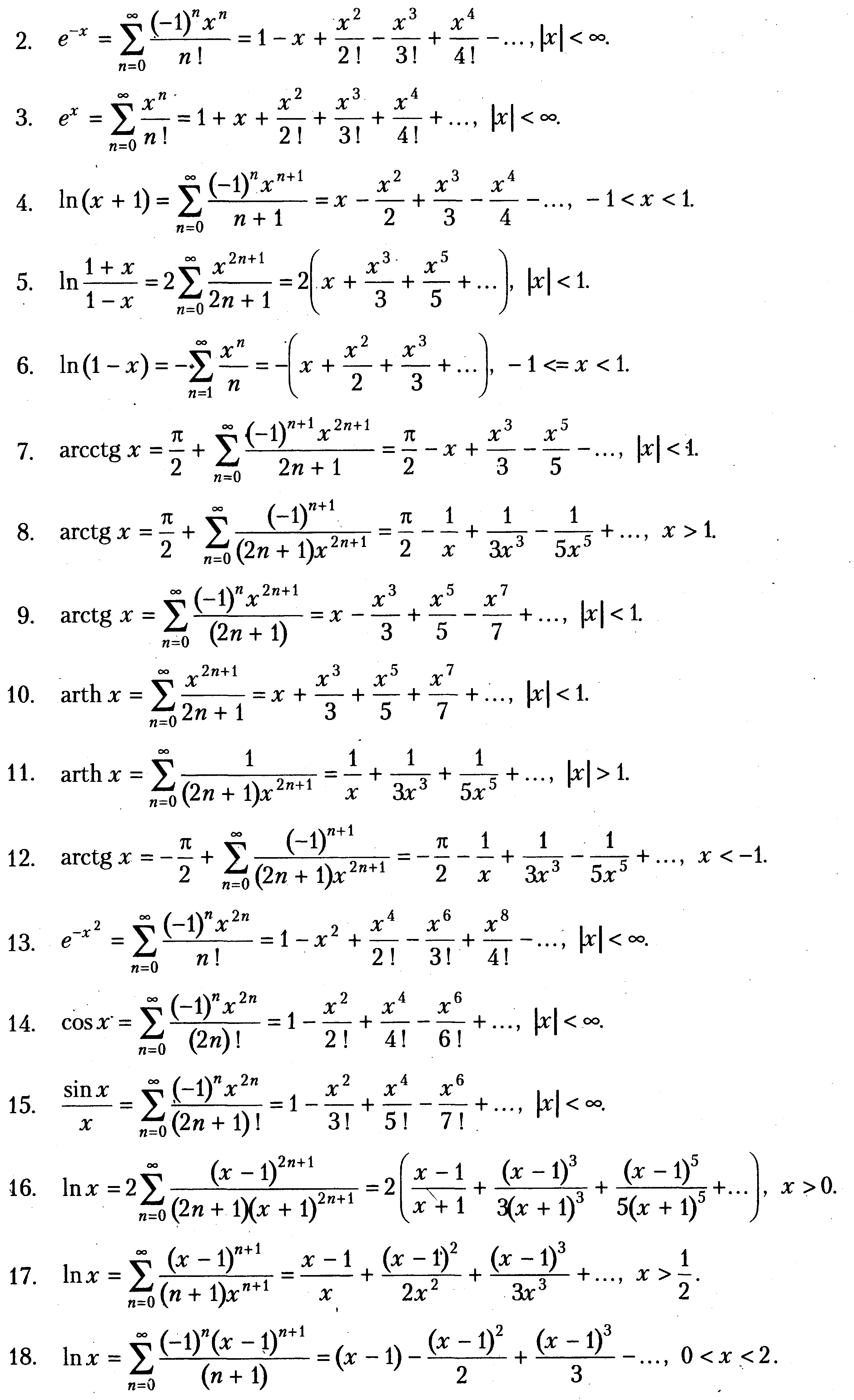
**19**. Ввести целое число N > 0. Найти значение выражения 1.1 – 1.2 + 1.3 – ... (N слагаемых, знаки чередуются). Условный оператор не использовать.

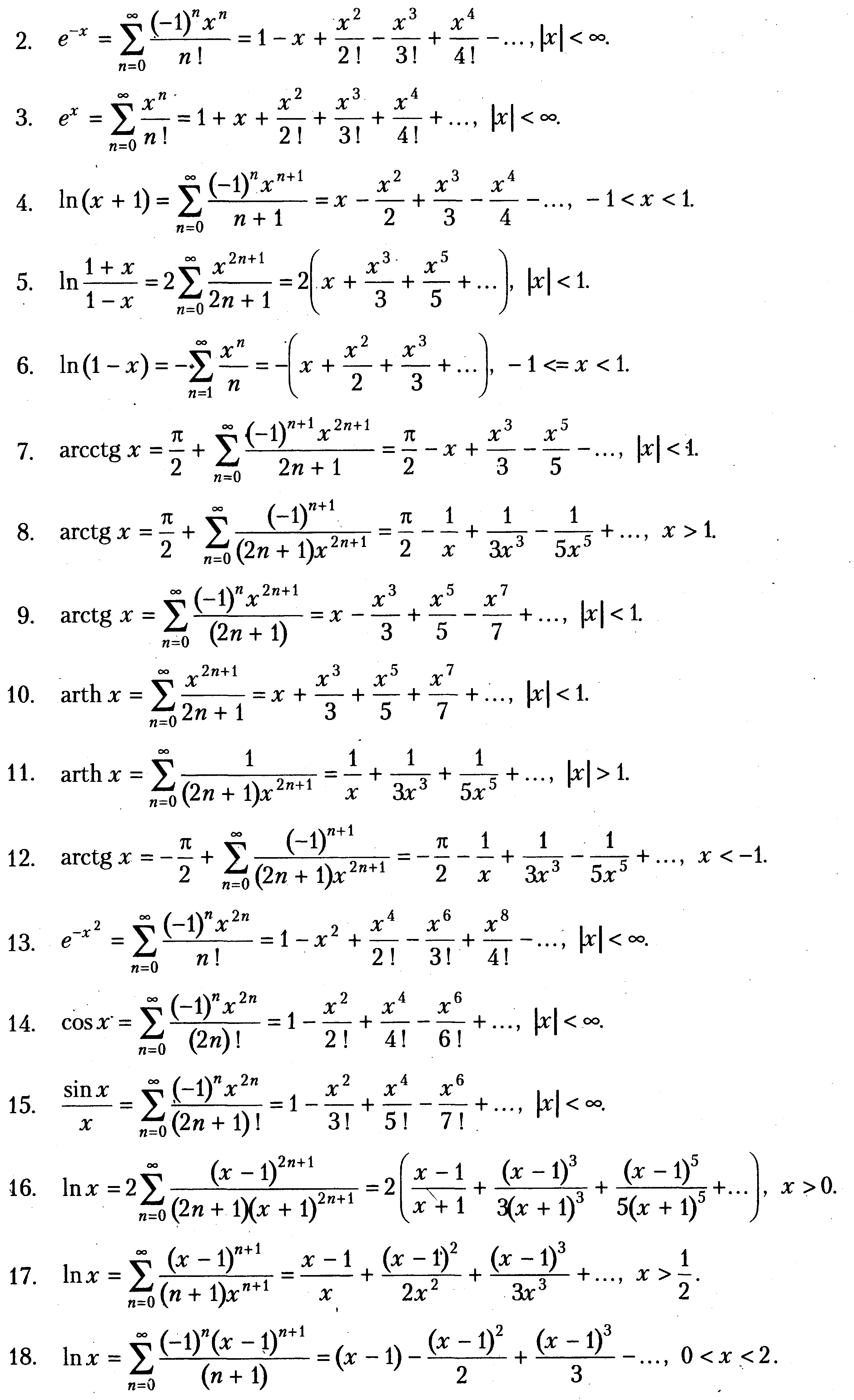
**20**. Ввести целое число N > 0. Среди цифр этого числа выделить только чётные, из которых составить другое число и вывести. Например, при N=3854972 ответом будет число 842.

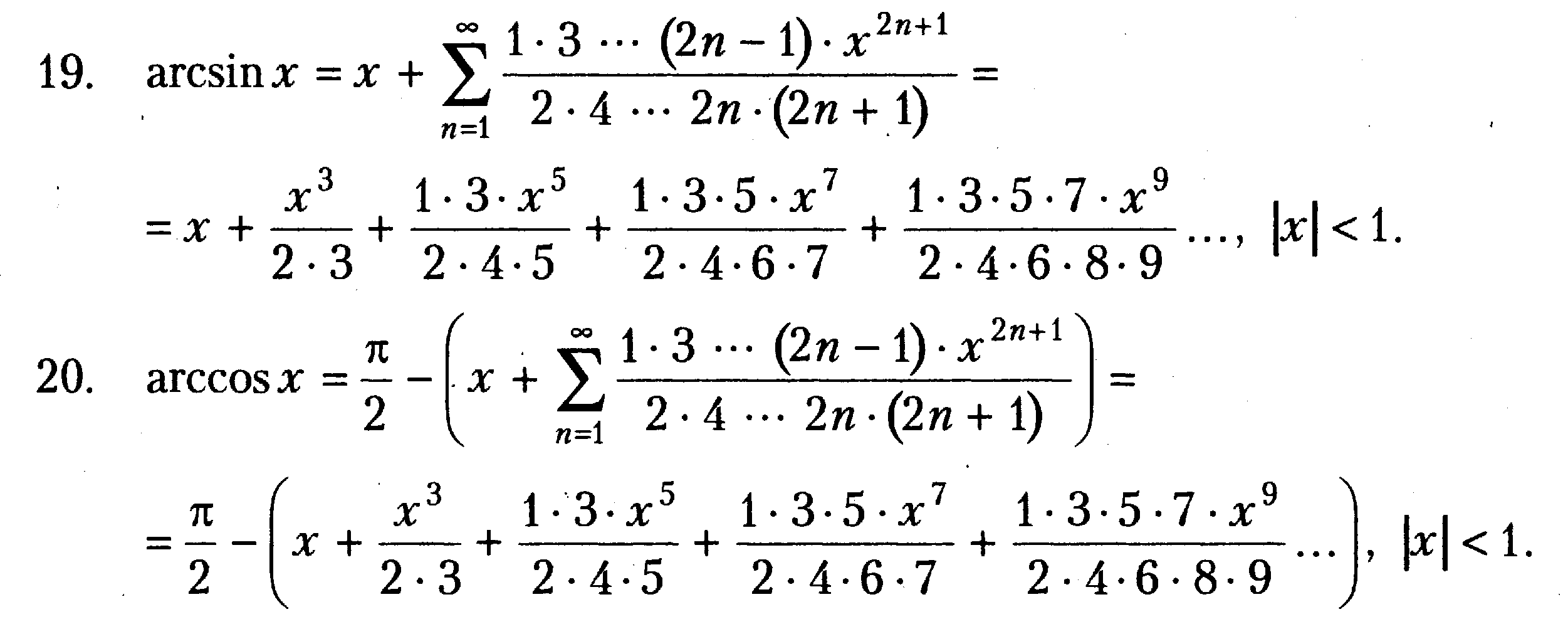
**Задание 3.** Вычислить и вывести на экран в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от хнач до хкон с шагом dx с точностью ε. Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значение аргумента, значение функции и количество просуммированных членов ряда.











1. **Оснащение работы**

Технические средства обучения:

* IBM – совместимый компьютер;

Электронные средства обучения:

* MS Visual Studio;
* MS Office.

1. **Основные теоретические сведения**

В языке C# для выбора одной из нескольких возможностей используются две конструкции - **if и switch**. Первую из них обычно называют альтернативным выбором, вторую - разбором случаев.

Синтаксис оператора if можно представить следующим образом:

if (выражение\_лог\_типа) оператор; //сокращенная форма

if (выражение\_лог\_типа>)оператор1; else оператор2;

В выражении должен получаться результат, имеющий логический тип. Если результатом выражения является истинное значение (True), то выполняется оператор, следующий за условием в скобках. Если результатом выражения является значение False и присутствует ключевое слово else, то выполнятся оператор, следующий за ключевым словом else. Если ключевое слово else отсутствует, то никакой оператор не выполняется. В предшествующем else операторе точка с запятой указывается. В общем случае ключевое слово else связывается с ближайшим ключевым словом if, которое еще не связано с ключевым словом else. Если вместо указанных операторов1,2 требуется выполнить несколько операторов используются операторные скобки {}.

|  |
| --- |
| [Блок-схема оператора if (показаны сокращённая и полная формы)](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=46) [[magnify-clip](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=46)](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=46)Блок-схема оператора if (показаны сокращённая и полная формы) |

**Пример.** Даны две переменные целого типа: A и B. Если их значения не равны, то присвоить каждой переменной большее из этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных A и B.

int A, B; string s;

Console.Write("Введите A=");

s = Console.ReadLine();

A = Convert.ToInt32(s);

Console.Write("Введите В=");

s = Console.ReadLine();

B = Convert.ToInt32(s);

if (A != B)

if (A > B) B = A;

else A = B;

else {A = 0; B = 0;}

Console.WriteLine("A={0} B={1}",A,B);

Console.ReadKey();

**Пример.** Даны a,b,c - коэффициенты квадратного уравнения. Получить корни уравнения.

double a,b,c,D,x1,x2; string s; //Описываем переменные

Console.Write("Введите A="); //Выдаём приглашение для ввода a

s = Console.ReadLine(); //Вводим строку s

a = Convert.ToDouble(s); //Преобразовываем строку в a

Console.Write("Введите B="); //Выдаём приглашение для ввода b

s = Console.ReadLine(); //Вводим строку s

b = Convert.ToDouble(s); //Преобразовываем строку в b

Console.Write("Введите C="); //Выдаём приглашение для ввода c

s = Console.ReadLine(); //Вводим строку s

c = Convert.ToDouble(s); //Преобразовываем строку в c

D = Math.Pow(b, 2) - 4 \* a \* c; //Определяем дискриминант

if (D > 0) // если дискриминант положительный будет два корня

{

x1 = (-b + Math.Sqrt(D)) / 2 / a;

x2 = (-b - Math.Sqrt(D)) / 2 / a;

Console.WriteLine("Два действительных корня x1={0}, x2={1}", x1, x2);

}

else

if (D==0) //если дискриминант равен нулю будет всего один корень

{

x1 = -b / 2 / a;

Console.WriteLine("Единственный действительный корень x={0}", x1);

}

else //все остальные случаи ,

{ //т.е. дискриминант отрицательный - решений нет

Console.WriteLine("Действительных корней нет");

}

Console.ReadKey();

Оператор выбора **switch** состоит из выражения (переключателя) и списка операторов, каждому из которых предшествует одна или более констант (они называются константами выбора) или ключевое слово **default**. Все константы выбора предваряются ключевым словом **case**, должны быть уникальными и иметь тип, совместимый с типом переключателя.

**Пример.**  Выдать введённое число в словесной интерпретации

int A; string s;

Console.Write("Введите A=");

s = Console.ReadLine();

A = Convert.ToInt32(s);

switch (A)

{

case 1: Console.WriteLine("Один"); break;

case 2: Console.WriteLine("Два"); break;

case 3: Console.WriteLine("Три"); break;

case 4: Console.WriteLine("Четыре"); break;

default: Console.WriteLine("Остальные числа"); break;

}

Управление передается оператору **case**, совпадающему со значением оператора **switch**. Оператор **switch** может включать любое количество экземпляров **case**, но два оператора **case** не могут иметь одинаковое значение. Выполнение текста оператора начинается с выбранного оператора и продолжается до тех пор, пока оператор **break** не передаст управление за пределы текста **case**. Оператор перехода, такой как **break**, требуется после каждого блока **case**, включая последний блок, вне зависимости от того, какой из двух операторов — **case** или **default** — там использован. Язык C# (в отличие от оператора switch в языке C++) не поддерживает неявное "проваливание" от одной подписи оператора **case** к другой, однако есть одно исключение. Исключением является случай, когда оператор case не имеет кода.

**Пример.** Случай с неявным проваливанием

int A; string s;

Console.Write("Введите A=");

s = Console.ReadLine();

A = Convert.ToInt32(s);

switch (A)

{

case 1: Console.WriteLine("Один"); break;

case 2: Console.WriteLine("Два"); break;

case 3: Console.WriteLine("Три"); break;

case 4: Console.WriteLine("Четыре"); break;

case 5:

case 6:

case 7:

case 8:

case 9:

case 10: Console.WriteLine("Число от 5 до 10"); break;

default: Console.WriteLine("Остальные числа"); break;

}

Console.ReadKey();

Если ни одно выражение case не совпадает со значением оператора **switch**, управление передается операторам, следующим за необязательной подписью **default**. Если подписи **default** нет, то управление передается за пределы оператора **switch**.

Цикл **for** повторно выполняет оператор или блок операторов, пока определенное выражение не примет значение **false**. Цикл **for** удобно использовать для итераций в массивах и для последовательной обработки. В следующем примере значение i печатается на экран, и затем увеличивается на 1 за каждое прохождение цикла. Рекомендуется использовать оператор **for** в тех случаях, когда заранее известно число повторений.

int i;

for (i = 1; i <= 5; i++)

Console.WriteLine(i); //тело цикла - всего один

//оператор и скобки {} не нужны

/\*напечатается

1

2

3

4

5 \*/

for (double j = 3.2+0.6; j > 2.1; j=j-0.3) //переменная j

Console.WriteLine(j);//локальна только для цикла

/\*напечатается

3.8

3.5

3.2

2.9

2.6

2.3

\*/

|  |
| --- |
| [Блок-схема для оператора for](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=33) [[magnify-clip](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=33)](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=33)Блок-схема для оператора for |

Оператор **for** повторно выполняет заключенный в цикл оператор (или несколько операторов) следующим образом:

* сначала вычисляется начальное значение параметра цикла;
* затем, пока условие получает значение true, оператор в теле цикла выполняется и затем происходит новое вычисление параметра цикла;
* когда условие получает значение false управление передается вне цикла.

Поскольку тестирование условного выражения осуществляется до выполнения цикла, **оператор for выполняется ноль или более раз**. Все выражения оператора for являются необязательными. Например, следующий оператор используется для создания бесконечного цикла:

for (;;)

{

// ...

}

Инструкция **break** прекращает выполнение цикла и управление передаётся за цикл. Инструкция **continue** выполняет пропускание операторов в теле цикла идущих ниже, осуществляется вычисление следующего параметра цикла и его дальнейшее повторение.

**Пример.**

double i;

for (i = 3.8; i > 2.1; i = i - 0.3)

{

Console.WriteLine(i);

if (i == 3.5) break;

}

Console.WriteLine(i);

/\* напечатается

3.8

3.5

3.5 \*/

for (i = 2; i < 7.2; i += 0.7)

{

if (i < 4.5) continue;

Console.WriteLine(i);

}

Console.WriteLine(i);

/\* напечатается

4.8

5.5

6.2

6.9

7.6 \*/

Можно изменять значение переменной внутри цикла

double i;

for (i = 3.8; i > 3; i = i - 0.3)

{

if (i == 3.5) i = 3.9;

Console.WriteLine(i);

}

Console.WriteLine(i);

/\*напечатается

3.8

3.9

3.6

3.3

3

2.7

хотя цикл должен был прекратиться раньше

из-за 3 этого не произошло из-за

погрешности в арифметических операциях

с действительным типом (3.00000000004)

\*/

**Пример.** Дано целое число N > 0. Найти сумму 11 + 22 + … + NN , не используя функций непосредственного возведения в степень.

int N, i,j, sum, s1; string s;

Console.Write("Введите N=");

s = Console.ReadLine();

N = Convert.ToInt32(s);

sum=0;

for (i=1;i<=N;i++)

{

s1=i;

for (j=1;j < i;j++) s1=s1\*i;

sum = sum + s1;

}

Console.WriteLine("Сумма ряда равна {0} ", sum);

Console.ReadKey();

Оператор **do** повторно выполняет оператор или блок операторов, заключенных в {}, пока определенное выражение не примет значение **false**. В следующем примере операторы цикла **do - while** выполняются до тех пор, пока пользователь не введёт значение из допустимого диапазона. Если в теле цикла всего один оператор, фигурные скобки ставить не нужно.

int N; string s;

do

{

Console.Write("Введите N в интервале 0..10 N=");

s = Console.ReadLine();

N = Convert.ToInt32(s);

}

while (N<0|| N>10);

Console.WriteLine("Введённое число равно {0} ", N);

Console.ReadKey();

|  |
| --- |
| [Блок-схема для оператора do-while](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=42) [[magnify-clip](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=42)](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=42)Блок-схема для оператора do-while |

Цикл **do-while** выполняется **как минимум один раз**, так как вычисление значения условного выражения выполняется после тела цикла. В любой точке блока **do-while** цикл можно разорвать с помощью оператора **break**. Для перехода непосредственно к оператору вычисления выражения **while** используется оператор **continue**; если выражение имеет значение **true**, выполнение продолжается в первом операторе цикла. Если выражение имеет значение false, выполнение продолжается в первом операторе после цикла **do-while**.

**Пример.** Дано целое число N > 0. С помощью операций деления нацело и взятия остатка от деления определить, имеются ли в записи числа N нечетные цифры.

int N; string s;bool f;

Console.Write("Введите целое положительное число N=");

s = Console.ReadLine();

N = Convert.ToInt32(s);

int n = N; //сохраним исходное значение

f=false;

do

{

if (N % 2 == 1) f = true;

N = N / 10;

} while (N != 0 && !f);

if (f) Console.WriteLine("Нечётные цифры в числе {0} есть", n);

else Console.WriteLine("Нечётных цифр в числе {0} нет", n);

Console.ReadKey();

Оператор **while** выполняет оператор или блок операторов, пока определенное выражение не примет значение **false**. Поскольку перед каждым выполнением цикла выражение **while** тестируется, цикл **while** выполняется от нуля до нескольких раз. Это отличает его от цикла **do**, который выполняется от одного до нескольких раз.

Цикл **while** может быть прерван оператором **break**, который передает управление за пределы цикла. Чтобы передать управление на следующую итерацию без выхода из цикла, используйте оператор **continue**.

**Пример.** Внесённая в банк сумма равна S, процентная ставка N годовых, определить с помощью цикла while на сколько лет нужно сделать размещение чтобы получить сумму F.

double S, F, N; string s; int i;

Console.Write("Введите начальную сумму S=");

s = Console.ReadLine();

S = Convert.ToDouble(s);

Console.Write("Введите процентную ставку годовых (%) N=");

s = Console.ReadLine();

N = Convert.ToDouble(s);

Console.Write("Введите желаемую сумму F=");

s = Console.ReadLine();

F = Convert.ToDouble(s);

i = 0; //счётчик годов

while (S < F)

{

S = S \* (1 + N / 100);

i++;

}

Console.WriteLine("Нужную сумму придётся ждать {0} лет, она составит {1} ", i,S);

Console.ReadKey();

## Пример. Написать программу, которая считывает введённое пользователем с клавиатуры целое число (использовать переменную целого типа) и выдает на экран дисплея сумму цифр этого числа.

### Решение

**Описание алгоритма:** поскольку программа должна суммировать цифры числа, предполагается выполнение циклических действий, так как в числе может быть несколько разрядов. Из всех операторов цикла лучше всего подходит цикл с постусловием, который выполняется как минимум один раз: это его свойство очень важно для нас, так как в любом числе хотя бы один разряд всегда есть.

В цикле будем суммировать разряды числа, на каждом шаге выделяя по одному разряду. Выделение самого младшего разряда числа (самой правой цифры числа) будем осуществлять нахождением остатка от деления на 10. Поскольку введённое число может быть отрицательным, то и остаток от деления нужно брать по модулю на каждом шаге. Далее необходимо отбросить самый младший разряд числа, для этого воспользуемся целочисленным делением на 10. Цикл будем повторять до тех пор, пока все разряды числа не закончатся, то есть в результате деления на 10 останется нуль.

|  |
| --- |
| [Блок-схема для примера 1](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=36) [[magnify-clip](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=36)](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=36)Блок-схема для примера 1 |

int n, sum; string s; //описываем необх. переменные

Console.Write("Введите целое число N=");

s = Console.ReadLine(); //вводим число

n = Convert.ToInt32(s); //преобразовываем к //целому типу

sum = 0; //обнуляем сумму перед циклом

do {

sum +=Math.Abs( n % 10); //прибавляем к //сумме самую правую цифру числа n

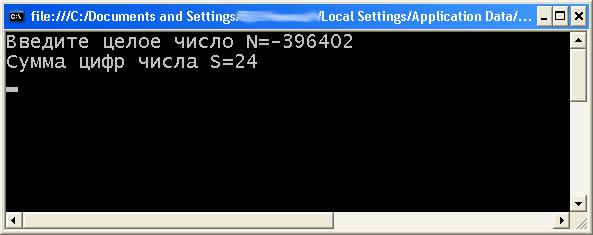
n /=10; //целочисленно делим n на 10,

//т.е. отбрасываем самую правую цифру

}while (n!=0); //повторяем пока не //закончатся все разряды числа n

Console.WriteLine("Сумма цифр числа S={0}",sum);

Console.ReadKey();

[](http://it.fitib.altstu.ru/neud/cs/index.php?action=viewimage&imn=35)

1. **Порядок выполнения работы**

1 Разработайте алгоритм решения задач в виде блок-схемы

2 Создайте консольное приложение в среде Microsoft Visual Studio

3 Осуществите ввод данных в разрабатываемых программах

4 Выполните задание согласно варианту

5 Осуществите вывод результата на консоль

6 Оформите отчет о работе

1. **Форма отчета о работе**

*Номер учебной группы \_\_\_\_\_\_\_\_*

*Фамилия, инициалы обучающегося \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Дата выполнения работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Тема работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Цель работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Задание: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Оснащение работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Результат выполнения работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

1. **Контрольные вопросы и задания**
2. Перечислите операторы ветвления?
3. Назовите операторы цикла?
4. Назовите опреаторы, относящиеся к операторам управления?

**Рекомендуемая литература**

**Албахари, Д.** C# 5.0. Справочник. Полное описание языка / Д. Албахари, Б. Албахари : [пер. с англ. Ю.Н.Артеменко]. М., 2013.

**Гриффитс, И.** Программирование на С# 5.0 / И. Гриффитс ; [пер. с англ. И.М. Райтмана]. М., 2014.

**Павловская, Т.А.** C#. Программирование на языке высокого уровня / Т.А. Павловская. СПб., 2015.

**Стиллмен, Э.** Изучаем С# / Э. Стиллмен, Дж. Грин. СПб., 2014.